DAA / AMES

~~R-FTCE~~ IN-07-CR

JOINT INSTITUTE FOR AERONAUTICS AND ACOUSTICS

90626

NCC 2-150

NCC 2-390

National Aeronautics and
Space Administration

Ames Research Center

Stanford University

JIAA-TR-77

262 P.

A COMPUTATIONAL STUDY OF THRUST AUGMENTING EJECTORS BASED ON A VISCOUS-INVISCID APPROACH

Thomas S. Lund, Domingo A. Tavella, and Leonard Roberts

Joint Institute for Aeronautics and Acoustics

Department of Aeronautics and Astronautics

Stanford University, Stanford, CA 94305

May 1987

(NASA-CR-181205) A COMPUTATIONAL STUDY OF THRUST AUGMENTING EJECTORS BASED ON A VISCOUS-INVISCID APPROACH (Stanford Univ.) 262 p Avail: NTIS HC A12/MF A01 CSCL 21E

N87-27666

Unclas

G3/07 0090626

JIAA-TR-77

A COMPUTATIONAL STUDY OF THRUST AUGMENTING EJECTORS BASED ON A VISCOUS-INVISCID APPROACH

Thomas S. Lund, Domingo A. Tavella, and Leonard Roberts

Joint Institute for Aeronautics and Acoustics
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305

May 1987

Abstract

Today's VSTOL aircraft designer is in need of an accurate theoretical model which can swiftly evaluate various ejector configurations. Previous attempts at developing such a model have been either over-simplified to the point of questionable accuracy, or so computationally expensive that optimization studies were not practical.

A viscous-inviscid interaction technique is advocated as both an efficient and accurate means of predicting the performance of two-dimensional thrust augmenting ejectors. The flow field is subdivided into a viscous region that contains the turbulent jet, and an inviscid region that contains the ambient fluid drawn into the device. The inviscid region is computed with a higher-order panel method, while an integral method is used for the description of the viscous part. The strong viscous-inviscid interaction present within the ejector is simulated in an iterative process where the two regions influence each other en route to a converged solution. This formulation retains much of the essential physics of the problem, but at the same time requires only a small amount of computing effort.

The model is applied to a variety of parametric and optimization studies involving ejectors having either one or two primary jets. The effects of nozzle placement, inlet and diffuser shape, free stream speed, and ejector length are investigated. The inlet shape for single-jet ejectors is optimized for various free stream speeds and Reynolds numbers. Optimal nozzle location and tilt are identified for various dual-jet ejector configurations.

In all cases, it is found that the dual-jet ejector out performs its single-jet counterpart. This fact is attributed to enhanced mixing due to an increase in the effective ejector length.

Acknowledgements

This work was supported by NASA grants NCC 2-150 and NCC 2-390 and was monitored by Mr. David Koenig of the Large-Scale Aerodynamics Branch, NASA-Ames Research Center.

Contents

List of Figures

Principal Symbols

a	Asymmetry factor in the dual-jet velocity profile
b	Jet excess velocity half-width
c_j	Velocity and pressure profile scale functions
C_p	Pressure coefficient
d	Ejector shroud thickness
H	Half of the ejector channel width
J	Primary jet momentum flux
k	Eddy-viscosity scaling coefficient
L	Ejector length
L_D	Diffuser length
M	Mach number
n	Normal direction, normal coordinate
p	Pressure
P_{atm}	Atmospheric pressure
R	Radius of curvature
R_e	Reynolds number
r	Distance from the field point to the point of integration on the body surface
s	Streamwise coordinate
T_0	Primary jet thrust
T_i	Induced thrust
t	Half of the jet nozzle width
u	Horizontal component of velocity, streamwise component of velocity
u_∞	Free stream velocity

v	Vertical component of velocity, stream-normal component of velocity
W	Half of the diffuser exit width
w_i	Weighting functions
x	Horizontal coordinate
x_j	Horizontal location of the jet nozzle
y	Vertical coordinate
y_j	Vertical location of the jet nozzle
α	Local surface angle, constant, jet tilt angle
β	Diffuser angle
Γ	Momentum equation operator
γ	Free stream speed parameter
ϵ	Residual error
ζ	Non-dimensional normal coordinate in the curved jet velocity profile
η	Non-dimensional vertical coordinate in the velocity profile
θ	Inlet lip rotation angle
κ	Surface curvature
Λ	Momentum equation operator less the turbulent diffusion term
λ	Velocity skewness parameter
μ	Doublet strength
ν_t	Eddy-viscosity coefficient
ξ	Local coordinate tangent to the body surface
ρ	Fluid density
σ	Source strength
τ	Turbulent shear stress
Φ	Velocity potential
ϕ	Thrust augmentation ratio, disturbance velocity potential

$(\tilde{\ })$ Quantity computed in the local coordinate system

$(\dot{\ })$ Derivative

$(\ddot{\ })$ Second derivative

$(\hat{\ })$ Approximate profile

$(\bar{\ })$ Quantity averaged across the ejector channel

Chapter 1

Introduction

1.1 Fundamental Physics Underlying Thrust Augmentation

A thrust augmenting ejector is a device capable of increasing the thrust produced by a propulsive jet nozzle through purely fluid mechanical means. The ejector consists of a high momentum primary jet that is exhausted into the confines of an aerodynamic shroud (see Figure 1.1). As the jet evolves, it entrains some of the ambient fluid contained within the ejector, thereby causing it to be swept downstream and through the ejector exit. The fluid lost to the jet entrainment is replaced by a secondary stream induced to flow in through the ejector inlet. As the secondary flow is accelerated around the leading edges of the ejector shroud, it lowers the local surface pressure in these regions. The resulting leading edge suctions create aerodynamic forces that have a large component in the direction of the primary nozzle thrust. These forces, together with the increased momentum flux of the primary nozzle due to the lowered pressure within the ejector, augment the force produced by the primary jet.

It is clear that the ability of the jet to entrain ambient fluid provides the mechanism of thrust augmentation. Most investigators refer to the effect of entrainment

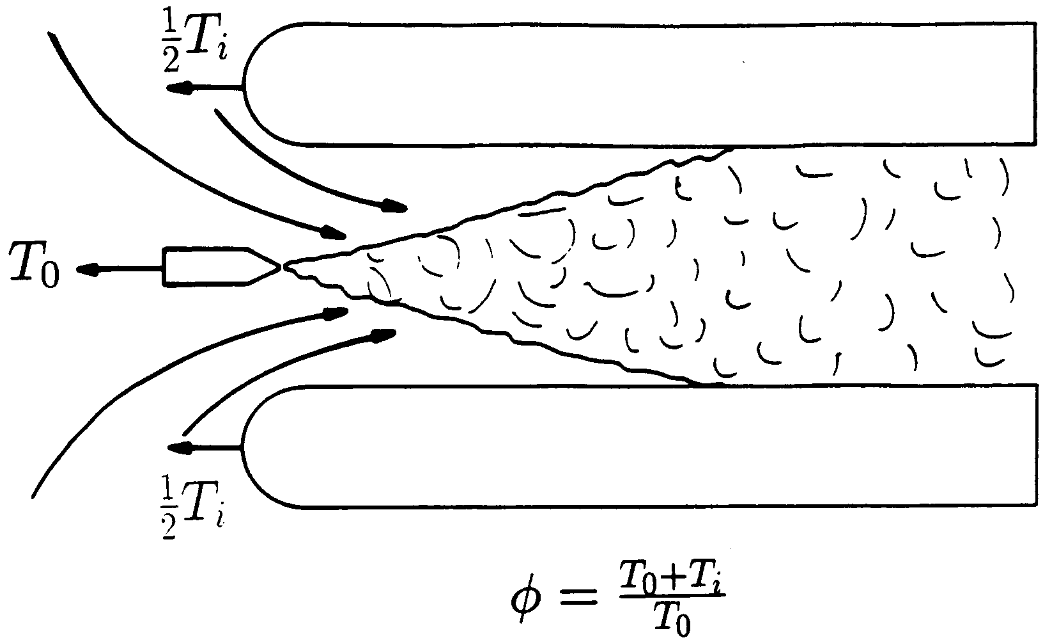


Figure 1.1: Thrust augmenting ejector concept

as "mixing" since the primary and entrained secondary flow become indistinguishable at the ejector exit station. The mixing that takes place within the ejector is due to a complex, turbulent process. While little is known about the details of the mixing process, the consequences of mixing are well understood. Ejectors perform optimally when the mixing process uniformly distributes the excess energy of the primary jet such that the exiting flow is at a thermodynamic state midway between the primary and secondary streams. Although to approach this limit of complete mixing is the goal of all ejector designs, the current lack of theoretical understanding of the mixing process has led to many configurations that are far from optimal. Theoretical models that realistically predict the mixing process are required to aid in the design of optimal ejectors.

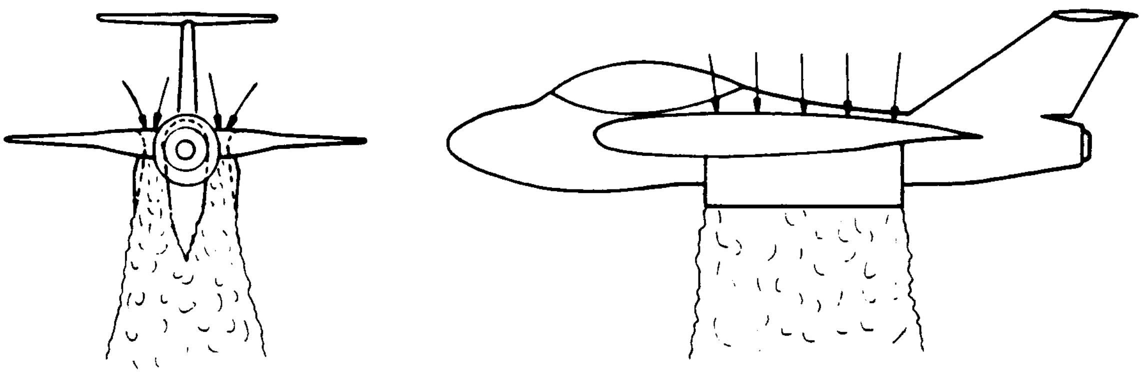


Figure 1.2: Ejector-fitted VSTOL aircraft

1.2 Application to VSTOL Aircraft Technology

The magnitude of the ejector effect is surprisingly large. Several investigators[1] have observed more than double the thrust produced by the primary jet alone. Because of its demonstrated potential as a thrust boosting device, the ejector has become an attractive component for advanced aerodynamics designs.

One important application of the thrust augmenting ejectors is found in vertical and short takeoff or landing (VSTOL) aircraft where there is a need for a large source of powered lift. In the ejector-powered vertical takeoff aircraft concept, the high pressure gas developed by the turbine engines is directed through a pair of ejectors mounted along the fuselage at the wing roots (see Figure 1.2). The ejectors boost the primary thrust to a level greater than the weight of the aircraft, thereby allowing it to rise vertically. Once sufficient altitude has been gained, the aircraft makes a conversion to forward flight by smoothly transferring the jet exhaust from the ejectors to the main horizontally thrusting nozzles. When the conversion is complete, the ejectors are covered over with movable doors to eliminate unnecessary drag. A vertical landing is achieved by repeating the takeoff procedure in reverse order.

1.3 Previous Work

Ejectors have been studied in connection with thrust augmentation since the mid 1920's. Today there exists a tremendous literature pertaining to ejector theory and performance. Comprehensive surveys of this work can be found in the review articles by Porter and Squyers[1] and Quinn[2]. The article by Porter and Squyers lists more than 1600 references. As a small subset of these, a selected number of important theoretical works are highlighted in this section.

The first theoretical study of ejectors was a control volume analysis given by von Karman[3]. In that analysis and those that followed[4,5,6] the ejector was treated as a black box where the conservation laws were required to hold only in a global sense between the entrance and exit stations. These control volume analyses have been quite useful in illustrating the importance of complete mixing as well as establishing theoretical limits to the maximum possible thrust augmentation.

In the control volume approach, the details of the ejector mixing process are collapsed into a single mixing efficiency parameter. This step allows simple analytic expressions for the thrust augmentation ratio to be determined. Unfortunately, the resulting expressions contain the mixing efficiency parameter as an unknown quantity. Most investigators have accepted this fact and have simply plotted performance curves with the mixing efficiency appearing as an undetermined parameter. In spite of the inability to connect the mixing efficiency to a particular ejector configuration, the control volume analyses are still useful in quantifying the importance of the degree of mixing. In addition, theoretical limits on the maximum possible thrust augmentation are established through the analyses by letting the mixing efficiency approach unity.

Without the ability to predict the ejector mixing process, the control volume analysis alone is not a powerful enough method to be used in conjunction with ejector design. The analysis can be supplemented with empirical information concerning the mixing efficiency, but this would require perhaps dubious extrapolations of the experimental data to investigate designs outside of the existing data base. A better alternative is to supplement the control volume analysis with a realistic theoretical model of the turbulent mixing process.

Much of the recent effort in ejector development has focused on developing realistic theoretical models of the mixing process. The basic approach in many contemporary works is to incorporate a turbulence model in the approximate solution to the Navier-Stokes equations which govern the ejector flow. If the turbulence model is reliable, the numerical simulations are able to predict the performance of an arbitrary ejector configuration. It is therefore possible to use these techniques to aid in ejector design.

A few investigators have attempted to model the ejector mixing process through a direct finite difference solution to the Navier-Stokes equations [7,8]. While the results have been encouraging, there is a practical problem in that these solutions require enormous amounts of computing time, even on the fastest class of computers. A single thin-layer Navier-Stokes calculation performed by Lasinski et al. [8], for example, took on the order of ten hours of processor time on a CDC 7600 machine. This sort of demand for computational power makes a full Navier Stokes simulation impractical for ejector design studies where hundreds of different configurations must be evaluated.

An alternative solution technique, known as the viscous-inviscid matching procedure, was first applied to the ejector mixing problem by Bevilaqua [9]. By making approximations locally and incorporating some of the known properties of jets, Bevilaqua was able to dramatically reduce the computational effort needed to model the ejector mixing process. Later improvements and extensions of this idea by Bevilaqua[10,11], Tavella[12,13], and Lund[14] have increased the accuracy and usefulness of the viscous-inviscid technique.

In the viscous-inviscid method, the flow field is divided into two separate regions. The turbulent flow consisting of the primary jet and mixed flow make up the viscous region, while the secondary, mainly irrotational flow makes up the inviscid region. Independent approximations are made in each region to simplify the problem while still resolving the important flow physics. The two regions are solved simultaneously in an iterative process that simulates the interaction between the jet and the ambient fluid. When the process converges, the flow variables are continuous at the juncture between the viscous and inviscid zones.

The efficient nature of the viscous-inviscid technique is attributed to its ability to utilize different approximations within the viscous and inviscid regions. The viscous-inviscid models developed to date have treated the inviscid region within a potential flow framework, and the viscous flow under a thin shear layer assumption. These are good local approximations that lead to a pair of relatively simple problems for which efficient solution techniques exist. The need to iterate between the two solutions does not become a great concern since each of the individual solution procedures are orders of magnitude more efficient than a Navier-Stokes solution.

Bevilaqua's original viscous-inviscid model[9] did not resolve the entire inviscid portion of the flow. In this first model, the inviscid secondary flow was assumed to be uniform at the ejector inlet station. Bevilaqua furthermore assumed that the jet could be modeled with a self-similar solution. These assumptions led to an extremely streamlined solution procedure that only required marching an initial value problem with a single unknown.

While the validity of the uniform secondary flow assumption as well as the self-similar jet solution could be disputed, Bevilaqua's original viscous-inviscid model illustrated a concept that could easily be improved to simulate the ejector flow field more accurately. In his two later works[10,11], Bevilaqua improved his original model by fully resolving the secondary flow with a combined panel/vortex lattice technique. The jet model was also improved by replacing the self-similar solution with a finite difference solution to the thin shear layer equations. These improved models were used successfully to predict the behavior of the ejector performance as a limited number of geometrical parameters were varied.

While Bevilaqua's improved viscous-inviscid technique represented the ejector flow physics quite realistically, the use of a finite difference solution in the viscous region reduced the overall efficiency of the method. In an effort to regain some of the lost efficiency while still maintaining an accurate solution, Tavella[12] developed an integral method for the viscous portion of the flow field. By making some reasonable assumptions regarding the shape of the velocity profile, Tavella was able to formulate a method that could generate essentially the same information as the finite difference calculation at a small fraction of the computational time. In a later work[13], Tavella combined his integral method with a conformal mapping solution for the inviscid flow. The resulting algorithm was nearly as efficient as Bevilaqua's original work, but produced a more realistic simulation of the ejector flow field.

One criticism of Tavella's viscous-inviscid model is that the conformal mapping technique used for the inviscid flow imposed a practical limitation on the shape of the ejector shroud. The method was restricted to shrouds that could be described by small perturbations to flat plates. An additional shortcoming of Tavella's model (and Bevilaqua's later models) was that the thickness of the jet was ignored in the inviscid solution. In both Tavella's and Bevilaqua's models, the jet was treated as a line of sinks along the ejector centerline, whose strengths were determined from the entrainment predicted by the viscous jet calculation. Accordingly, the flow variables were matched at the ejector channel centerline and not the viscous-inviscid interface.

1.4 Present Work

The objective of the present work is to improve upon the existing viscous-inviscid matching techniques in order to create an accurate and robust model that is efficient enough to be used as an ejector design tool. The improvements entail both a synthesis and extension of the existing methods. These may be summarized as follows:

1. Use a higher-order panel method for the inviscid flow so that arbitrary shroud shapes can be studied.

2. Combine the higher-order panel method with the integral method of solution for the viscous flow.

3. Take the jet thickness into account in the inviscid solution and thereby match the flow variables at the viscous-inviscid boundary as opposed to the ejector channel centerline.

4. Extend the integral method for the case of an ejector with two primary jets.

5. Develop a second complete viscous-inviscid model for a dual-jet ejector.

A secondary objective of this work is to use the improved viscous-inviscid models to learn more about the performance characteristics of ejectors. In particular the aim is to:

1. Quantify the impact on performance when several ejector geometrical parameters are systematically varied.
2. Quantitatively compare the performance of a dual-jet ejector with a single-jet ejector for a large range of configurations and operating conditions.
3. Use the models in some practical design problems to optimize the geometry for several different operating conditions.

1.4.1 Theoretical Framework

In the present work, several simplifying assumptions are made at the outset. These assumptions are designed to limit the scope of the problem while not being so restrictive that the analysis is of limited value. The assumptions may be listed as follows:

1. The mean flow is assumed to be steady.
2. The flow is assumed to be two-dimensional.
3. The flow is assumed to be incompressible.

The first assumption limits the analysis to steady flow ejectors. Most ejector designs are of this type, even though pulsed flow ejectors [15,16,17] have shown to produce more efficient mixing.

In the second assumption, the ejector flow field is idealized as being two-dimensional. This assumption is a reasonable approximation since many ejector designs have moderately large aspect ratios. Excluding the end regions, the bulk of the flow in real ejectors should behave as if it were two-dimensional. The three-dimensional effects that occur in the corner regions almost always lower the ejector performance.

Hence, the two-dimensional calculations can be considered to be an upper bound for the performance values that will be found in practice.

The third assumption limits the analysis to incompressible flow. This assumption is perhaps the most restrictive, since most of the modern ejectors are designed to operate with primary jet exit Mach numbers high enough to induce compressibility effects. The incompressible flow assumption can be viewed as a simplification necessary to limit the scope of the analysis in the first step towards producing a general, efficient ejector model. Once a methodology is established and tested for incompressible flow, it should be relatively simple to extend the model to include compressibility effects. In any event, the results of the present analysis are expected to produce a reasonable estimate of ejector performance for moderate primary jet Mach numbers. A more precise analysis of the applicability of the present model to compressible flows is given in Appendix A.

1.5 Overview

The thesis is divided into eight chapters. Chapter 2 introduces a control volume analysis that illustrates some of the basic properties of ejectors. Chapter 3 discusses the viscous-inviscid approach as it applies to the ejector problem. In Chapter 4 the higher-order panel method used for the inviscid solution is presented. Chapter 5 contains a derivation of the integral methods for both single and dual-jet ejectors. The matching procedure used to drive the iteration between the viscous and inviscid solutions is presented in Chapter 6. Chapter 7 contains the results of both the parametric and optimization studies. Finally, a summary and some of the major conclusions are listed in Chapter 8.

Chapter 2

Classical Analysis

Theoretical analyses of ejectors can be grouped into two categories: (1) approximate solutions to the equations of motion and (2) control volume analysis. In the first of these two categories, the flow variables are determined at each point within the ejector by employing a numerical technique to solve the appropriate equations of motion. This type of analysis resolves the details of the mixing process and can therefore be used to learn more about the physics of thrust augmentation. Although the numerical simulation approach yields a wealth of information about the ejector flow field, it is difficult to implement. The control volume approach, on the other hand, is easy to implement and gives analytical results that provide some useful information about the global properties of ejectors. Before the advent of computers, control volume approaches were used almost exclusively to analyze ejectors. Today, the results of these classical analyses are still useful in validating modern numerical simulations. A control volume analysis is presented here to provide some insight to the properties of ejectors and to be used later to support the results of the viscous-inviscid numerical simulation.

2.1 Control Volume Analysis

In the control volume approach the ejector is treated at a black box where conservation of mass, momentum, and energy are required to hold only between the inlet and exit stations. Global quantities such as the thrust augmentation ratio are determined without regard to the details of the mixing taking place within the ejector. The analysis is incomplete in this regard, and the degree of mixing is input as a known parameter. In spite of the need to specify a measure of mixing efficiency, the control volume analysis can show how the inlet velocity non-uniformity, free stream speed, and the addition of a diffuser affect the thrust augmentation ratio.

Control volume approaches have been widely used in the past. The original paper by von Karman[3] for incompressible flow ejectors without diffusers has been followed by several extensions to compressible flow, diffusers, and forward speed[4,5,6]. In this chapter the existing results are unified into a single analysis valid for incompressible flow.

Control volume analyses require the mixing process to take place either at constant pressure or for constant area. The analysis which is given here is for constant area mixing. The flow is also assumed to be incompressible and one-dimensional.

The equations of motion for incompressible flow are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.1}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} = \frac{1}{\rho}\frac{\partial \tau}{\partial y} \tag{2.2}$$

where τ is the turbulent shear stress. The continuity equation is used to rewrite the momentum equation as

$$\frac{\partial}{\partial x}\left(u^2 + \frac{1}{\rho}p\right) + \frac{\partial uv}{\partial y} = \frac{1}{\rho}\frac{\partial \tau}{\partial y} \tag{2.3}$$

Equations (2.1) and (2.3) are now integrated across the ejector channel of constant half-width H. The configurations are assumed to be symmetric so that it is sufficient to consider only the upper half-plane.

$$\frac{\partial}{\partial x}\int_0^H u dy + v(H) - v(0) = 0 \tag{2.4}$$

$$\frac{\partial}{\partial x}\int_0^H \left(u^2 + \frac{1}{\rho}p\right) dy + u(H)v(H) - u(0)v(0) = \frac{1}{\rho}(\tau(H) - \tau(0)) \tag{2.5}$$

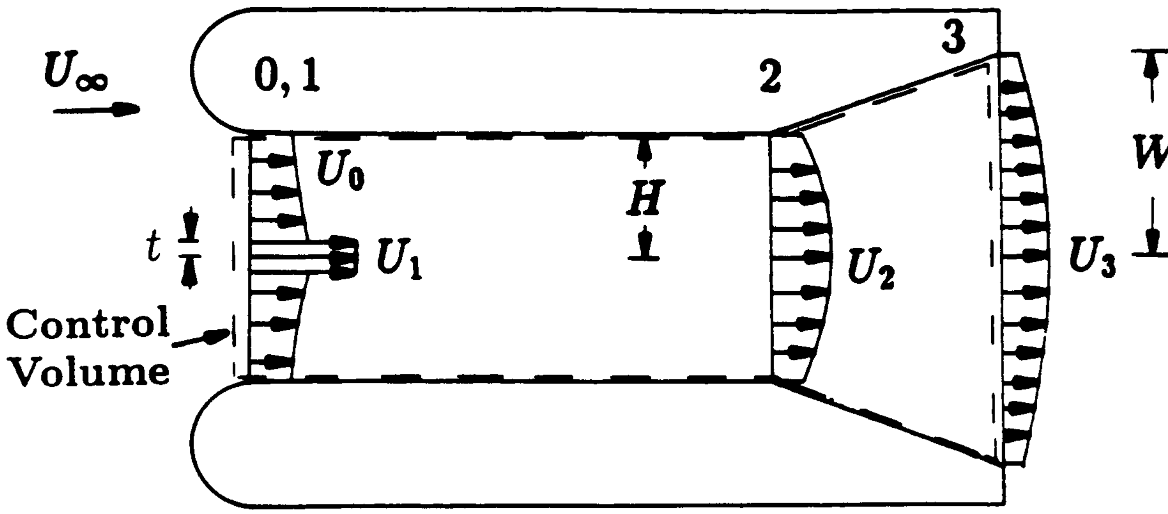


Figure 2.1: Ejector control volume schematic

At the channel centerline, both the vertical component of velocity v and the shear stress τ vanish by symmetry. At the channel wall, the v component of velocity again vanishes and the shear stress may be neglected (the skin friction may be incorporated later through an appropriate loss factor). With these ideas the conservation integrals become

$$\int_0^H u dy = \text{const} \tag{2.6}$$

$$\int_0^H \left(u^2 + \frac{1}{\rho}p\right) dy = \text{const} \tag{2.7}$$

Consider the schematic of the ejector shown in Figure 2.1. The conservation integrals are applied between stations 0 and 2 to give

$$\int_0^t u_1 dy + \int_t^H u_0 dy = \int_0^H u_2 dy \tag{2.8}$$

$$\int_0^t u_1^2 dy + \int_t^H u_0^2 dy + \int_0^H \frac{1}{\rho} p_0 dy = \int_0^H u_2^2 dy + \int_0^H \frac{1}{\rho} p_2 dy \tag{2.9}$$

Define the average properties

$$\bar{u} = \frac{1}{H}\int_0^H u dy \tag{2.10}$$

$$\bar{p} = \frac{1}{H}\int_0^H p dy \tag{2.11}$$

and the velocity skewness parameter

$$\lambda = \frac{\frac{1}{H}\int_0^H u^2 dy}{\left[\frac{1}{H}\int_0^H u dy\right]^2} \tag{2.12}$$

The velocity skewness parameter can be found in many of the previous control volume approaches[3,6,5]. It provides a measure of the flow non-uniformity. A uniform flow has a skewness value of 1, while increasingly non-uniform flows have higher values of the skewness parameter. In physical terms, the skewness parameter for incompressible flow is proportional to the ratio of the momentum flux to the square of the mass flux. Thus a flow with a skewness factor greater than unity contains more momentum than does a uniform flow with the same mass flux.

Using the above definitions, Eqs. (2.8) and (2.9) may be written in terms of the averaged quantities

$$\bar{u}_1 \frac{t}{H} + \left(1 - \frac{t}{H}\right)\bar{u}_0 = \bar{u}_2 \tag{2.13}$$

$$\lambda_1 \bar{u}_1^2 \frac{t}{H} + \left(1 - \frac{t}{H}\right)\lambda_0 \bar{u}_0^2 + \left(1 - \frac{t}{H}\right)\frac{1}{\rho}\bar{p}_0 = \lambda_2 \bar{u}_2^2 + \frac{1}{\rho}\bar{p}_2 \tag{2.14}$$

As an approximation, the primary nozzle is modeled as a point source of momentum. That is, the dimension of the nozzle is allowed to become arbitrarily small while the momentum flux is held fixed. The exit velocity is required to be unbounded in this instance in such a way that the nozzle is a singular point of finite momentum flux but with no associated mass flux.

The point source approximation is introduced by letting $\frac{t}{H} \to 0$ while $\bar{u}_1 \to \infty$ such that $\lambda_1 \bar{u}_1^2 \frac{t}{H} \to T_0/\rho H$. Then the above equations become

$$\bar{u}_0 = \bar{u}_2 \tag{2.15}$$

$$\frac{T_0}{\rho H} + \lambda_0 \bar{u}_0^2 + \frac{1}{\rho}\bar{p}_0 = \lambda_2 \bar{u}_2^2 + \frac{1}{\rho}\bar{p}_2 \tag{2.16}$$

Bernoulli's equation is averaged across the channel to give

$$\bar{p} = \bar{p}_T - 1/2\rho(\lambda \bar{u}^2 - u_\infty^2) \tag{2.17}$$

Assume that only a negligible amount of mixing takes place in the diffuser. Under this assumption, the flow between stations 2 and 3 is isentropic. The Bernoulli equation can therefore be applied between these two stations to give

$$\bar{p}_2 - \bar{p}_3 = -1/2\rho(\lambda_2\bar{u}_2^2 - \lambda_3\bar{u}_3^2) \tag{2.18}$$

The exit pressure must be equal to the atmospheric value. Thus $\bar{p}_3 = p_{atm}$. Conservation of mass requires $\bar{u}_3 = (H/W)\bar{u}_2$. The skewness factors λ_2 and λ_3 are equal since the process is isentropic. Making use of these results, the above equation becomes

$$\bar{p}_2 - p_{atm} = -1/2\rho\lambda_2\bar{u}_2^2\left[1 - \left(\frac{H}{W}\right)^2\right] \tag{2.19}$$

Bernoulli's equation may also be applied to the inviscid portion of the inlet flow to give

$$\bar{p}_0 - p_{atm} = -1/2\rho(\lambda_0\bar{u}_0^2 - u_\infty^2) \tag{2.20}$$

Equations (2.15), (2.16), (2.19), and (2.20) are now combined to yield

$$\left(\frac{\bar{u}_2}{u_\infty}\right)^2 = \frac{2+\gamma^2}{\gamma^2\lambda_2\left[1 + \left(\frac{H}{W}\right)^2 - \frac{\lambda_0}{\lambda_2}\right]} \tag{2.21}$$

where a measure of the free stream speed, γ, is defined as

$$\gamma^2 = \frac{\rho u_\infty^2 H}{T_0} \tag{2.22}$$

The thrust augmentation ratio is defined as

$$\phi = \frac{\int_0^W u_3(u_3 - u_\infty)dy}{\int_0^t u_1(u_1 - u_\infty)dy} \tag{2.23}$$

As before, let $\frac{t}{H} \to 0$. Then the above equation may be written as

$$\phi = \frac{(\lambda_3\bar{u}_3^2 - u_\infty\bar{u}_3)\frac{W}{H}}{\frac{T_0}{\rho H}} \tag{2.24}$$

Again assume $\lambda_3 = \lambda_2$. Then using the mass conservation relation, $\bar{u}_3 = (H/W)\bar{u}_2$, as well as the definition of the free speed parameter given in Eq. (2.22), the above relation becomes

$$\phi = \gamma^2\left[\lambda_2\left(\frac{\bar{u}_2}{u_\infty}\right)^2\frac{W}{H} - \left(\frac{\bar{u}_2}{u_\infty}\right)\right] \tag{2.25}$$

Equation (2.21) is used to give the final result

$$\phi = \frac{2+\gamma^2}{\left[1+\left(\frac{H}{W}\right)^2-\frac{\lambda_0}{\lambda_2}\right]}\frac{H}{W} - \gamma\sqrt{\frac{2+\gamma^2}{\lambda_2\left[1+\left(\frac{H}{W}\right)^2-\frac{\lambda_0}{\lambda_2}\right]}} \tag{2.26}$$

The velocity skewness parameters λ_0 and λ_2 can not be determined by the control volume analysis. The skewness parameter λ_0 represents the non-uniformity of the secondary flow. In many cases the secondary flow is nearly uniform and $\lambda_0 \simeq 1$. The skewness parameter λ_2 represents the degree of mixing of the primary and secondary streams within the constant area portion of the ejector channel. A value of $\lambda_2 = 1$ represents complete mixing where the flow exiting from the ejector is uniform and is at a thermodynamic state midway between the primary and secondary streams. Values of λ_2 are typically larger than 1.2. It is observed experimentally that λ_2 varies inversely with the ejector length. This is due to the fact that a longer ejector gives the flow more time to mix. The use of multiple primary jets or hypermixing nozzles should therefore also reduce λ_2.

Without prior knowledge of the velocity skewness factors, the control volume analysis can not be used to predict the performance of a particular configuration. The analysis is still useful, however, since it can be used to show how the performance will vary with these parameters. In addition, the effects of the free stream speed as well as the effects of a diffuser may be investigated. It is most instructive to isolate three special cases. These are:

1. Effects of the velocity skewness parameters; given $\gamma = 0,\ \frac{H}{W} = 1$:

$$\phi = \frac{2}{2-\frac{\lambda_0}{\lambda_2}} \tag{2.27}$$

2. Effects of a diffuser; given $\gamma = 0,\ \lambda_0 = 1$:

$$\phi = \frac{2\frac{W}{H}}{1+\left(\frac{W}{H}\right)^2\left[1-\frac{1}{\lambda_2}\right]} \tag{2.28}$$

3. Effects of a free stream; given $\frac{H}{W} = 1,\ \lambda_0 = 1$:

$$\phi = \frac{2+\gamma^2}{2-\frac{1}{\lambda_2}} - \gamma\sqrt{\frac{2+\gamma^2}{\lambda_2\left(2-\frac{1}{\lambda_2}\right)}} \tag{2.29}$$

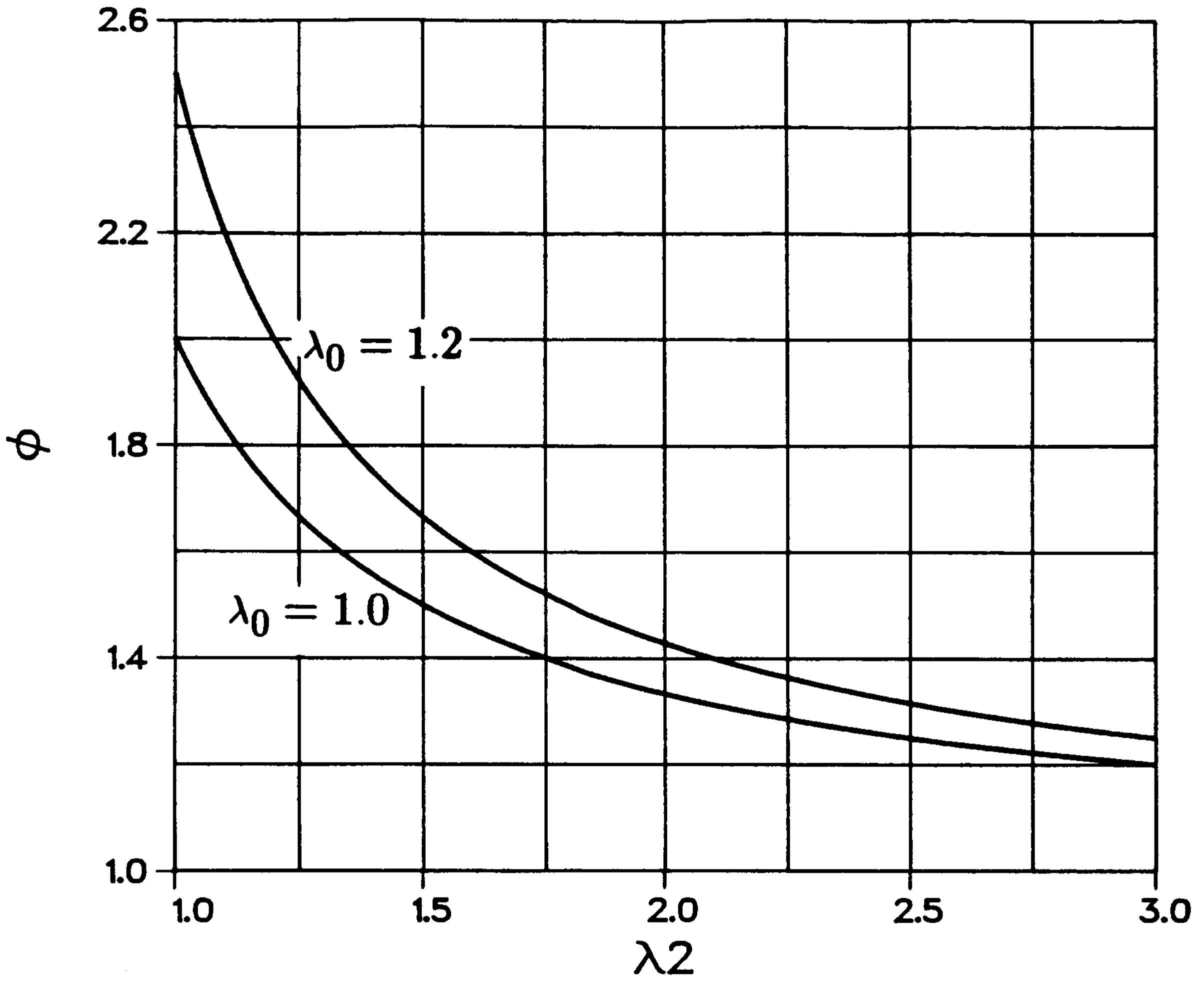


Figure 2.2: Effects of the velocity skewness parameters. $\gamma = 0$, $\frac{H}{W} = 1$.

2.2 Results

The effects of the velocity skewness parameters are shown in Figure 2.2, where Eq. (2.27) is plotted. As anticipated, the performance decreases with increasing λ_2. The performance is seen to increase with increasing λ_0. Thus the ejector performs better if the secondary flow is other than uniform. Showing that this is the case was the intent of von Karman's original paper[3]. Note that for nearly complete mixing ($\lambda_2 = 1$), the performance is significantly improved by secondary flow non-uniformity. As the mixing efficiency drops ($\lambda_2 > 1$), the secondary flow non-uniformity has a smaller impact on the performance. In summary, it is best

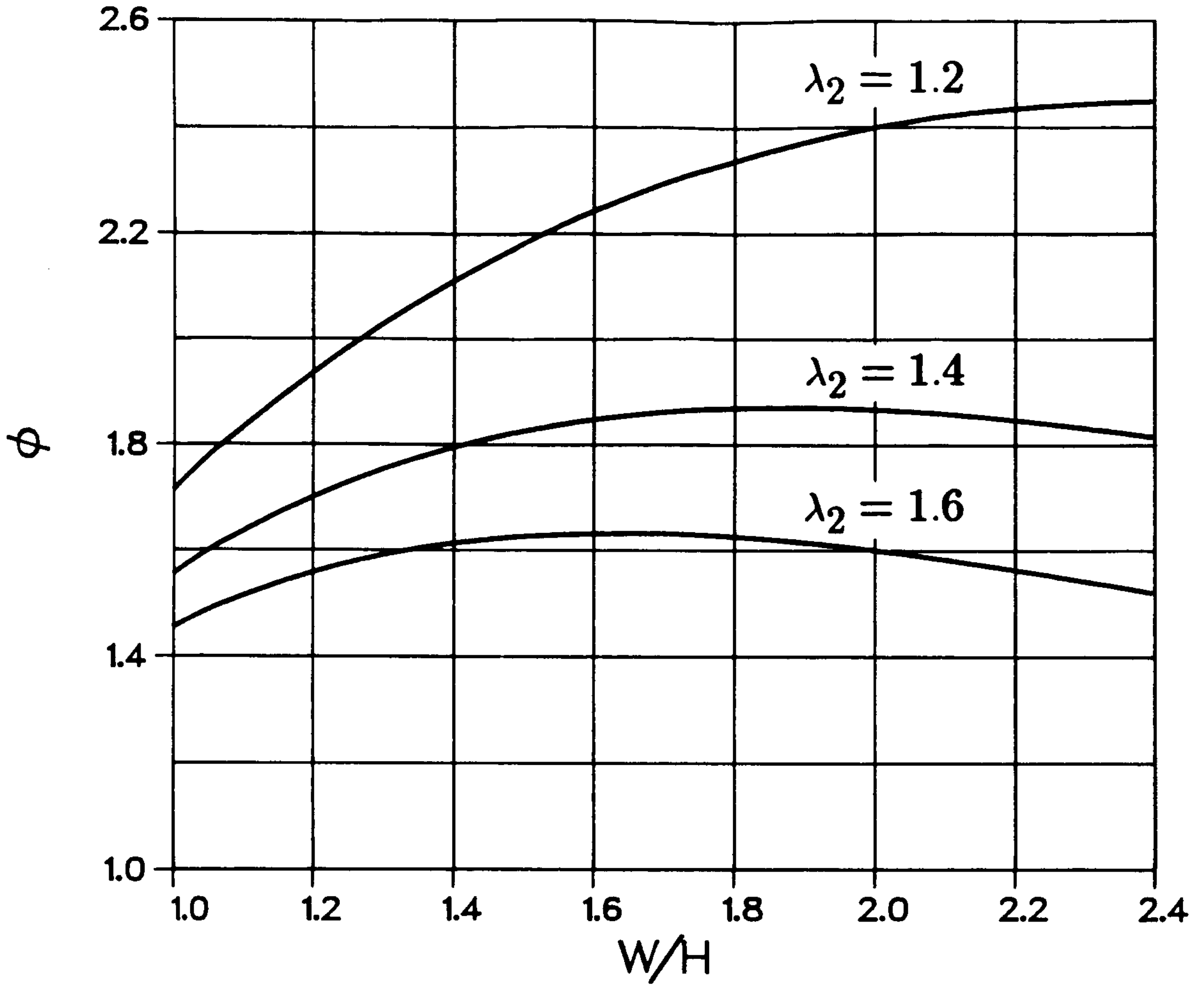


Figure 2.3: Effects of a diffuser. $\gamma = 0$, $\lambda_0 = 1$

to have a high degree of non-uniformity at the ejector inlet and nearly uniform conditions at the ejector exit.

The effects of a diffuser predicted by Eq. (2.28) are shown in Figure 2.3. It is evident that a diffuser is most beneficial if the flow entering the diffuser is close to being completely mixed (λ_2 near 1). The advantage of having the flow more nearly mixed is increasingly pronounced as the diffuser area ratio becomes large. This is an important result since it shows that ejectors that employ multiple primary jets or hypermixing nozzles in an effort to enhance the mixing process will benefit most from the addition of a diffuser.

Note that for each exit velocity skewness parameter, there is an optimal diffuser

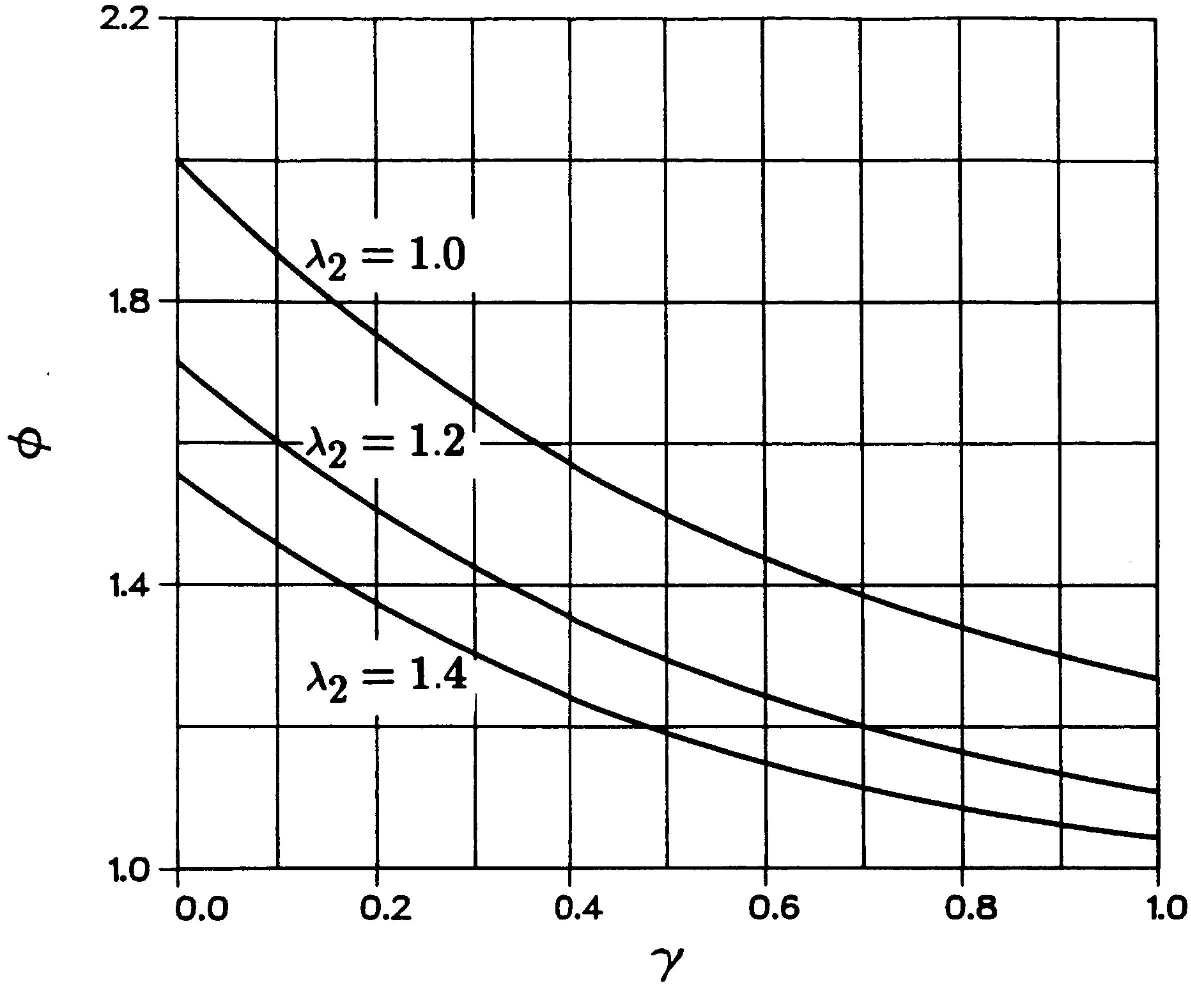


Figure 2.4: Effects of the free stream speed. $\frac{H}{W} = 1$, $\lambda_0 = 1$

area ratio. At this point the pressure drag associated with the diffuser starts to outweigh the increase in performance due to the lowered inlet pressure. The optimal diffuser area ratio is found from Eq. (2.28) to be

$$\left(\frac{W}{H}\right)_{max} = \frac{1}{\sqrt{1 - \frac{1}{\lambda_2}}} \tag{2.30}$$

In summary, the performance of an ejector with a diffuser is again best when the exiting flow is nearly mixed.

The effects of the free stream speed predicted by Eq. (2.29) are shown in Figure 2.4. The performance decreases monotonically with increasing free stream speed as a result of increasing ram drag. As in the other cases, the performance is always best when λ_2 is close to 1.

2.3 Conclusions

The control volume analysis has produced several interesting results. The analysis showed that the performance is always best as λ_2 approaches unity and as λ_0 departs from unity. It was shown that an optimal diffuser area ratio exists for each value of λ_2. The performance was also shown to decrease with the free stream speed parameter γ.

While these results are both interesting and instructive, they are limited by the need to prescribe the degree of mixing through the parameter λ_2. Because of this limitation, it is not possible to use the control volume analysis to predict the performance of a particular configuration. Since the focus of this work is to develop a model capable of such predictions, the control volume analysis must be supplemented with a realistic model of the ejector mixing process. The next several chapters describe a numerical simulation technique that is developed to provide the information necessary to determine the degree of mixing achieved by any given ejector configuration.

Chapter 3

Viscous-Inviscid Approach

In the viscous-inviscid approach, the field is divided into two separate regions or "zones" that contain flows of differing character. Regions of the flow that are not affected by viscous or turbulent stresses comprise the inviscid zone, while regions that contain significant fluid shear, such as boundary layers, jets, and wakes make up the viscous zone. Approximations are made independently in each zone to simplify the problem while still resolving the important flow characteristics. The independent approximations lead to two different sets of simplified equations, each of which is valid only in its respective region. The two zones are solved simultaneously in an iterative matching process which assures that the solution is continuous at the zonal interface. The converged solution is identical to a solution produced by a single set of equations that are valid for the whole domain, but is produced with a fraction of the computational effort.

3.1 Previous Work

The viscous-inviscid technique has been successfully used in the past to solve a variety of complex flows. Boundary layers which develop in turbomachinery [18,19], and wing-body junctures [20,21,22] have been treated with the viscous-inviscid method, as have flows involving shock-boundary layer interactions[23,24,25]. A large body of literature exists for viscous-inviscid methods applied to separated regions in both steady [26,27,28,29,30,31,32] and unsteady[33,34,35] flows. Confined jets and thrust augmentor configurations have also been modeled with these methods [9,10,12,13].

Perhaps the most familiar application of the viscous-inviscid method is the usual procedure for calculating boundary layers in aerodynamic flows. In an airfoil problem, the viscous zone is made up of a thin layer near the surface, while the inviscid zone covers the rest of the field. Typically, a potential flow method is used for the inviscid zone, while von Karman's integral method is used in the viscous zone. The inviscid solution provides the surface pressure distribution needed as a boundary condition to solve the boundary layer equations. The effect of the viscous region on the inviscid flow is then taken into account by increasing the thickness of the airfoil to simulate the displacement effect of the boundary layer. The thickness correction allows an improved inviscid solution to be generated. The new pressure distribution can then be used to compute yet another viscous flow, and so on. In principle, the cycle can be repeated until some desired degree of convergence is obtained. In practice, the interaction between the boundary layer on an airfoil and the surrounding inviscid stream is weak enough that only one iteration is needed to accurately match the two solutions. Other viscous-inviscid problems, such as a boundary layer with a separation bubble, involve a higher degree of interaction, and several cycles are necessary in order to match the solutions together.

The most attractive feature of the viscous-inviscid procedure is that it gives an accurate solution at a very modest computational cost. This advantage is attributed to the ability to solve a different set of equations in each of the two zones. Approximations are made locally, so that negligible terms are pruned where they are not needed. For example, in the viscous zone of the airfoil problem, streamwise diffusion is neglected and the velocity normal to the surface is assumed to be of higher order. These assumptions reduce the Navier-Stokes equations to the boundary layer equations. Being parabolic, the boundary layer equations are much easier to solve than the elliptic Navier-Stokes equations. The flow outside of the airfoil boundary layer is assumed to be inviscid and, if there are no strong shocks, irrotational. For purely subsonic flow, these assumptions allow the Navier-Stokes equations to be reduced to Laplaces equation, which again is much easier to solve than the Navier-Stokes equations. Thus, the viscous-inviscid formulation reduces the problem of solving the Navier-Stokes over the entire domain to that of solving two much simpler problems. The price for making this simplification is that the two solutions must be iteratively matched together. This is not a great concern, however, since convergence is often obtained in a few cycles, and the entire matching process is still significantly faster than solving the Navier-Stokes equations.

A further advantage of the zonal approach is that it is many times easier to implement. In some cases, one portion of the flow field may be simple enough to be described by an analytic solution. The other region may require a numerical solution, but the two may still be matched together to give the desired result. In other cases, only one portion of the flow field may need a computational grid. This can eliminate problems associated with generating a grid to fit a complicated geometry.

3.2 Ejector Problem

The viscous-inviscid approach is a natural choice for an ejector flow field since it contains well-defined regions of viscous and inviscid flow. The entrained secondary flow forms the inviscid zone, while the turbulent jet and boundary layers on the shroud walls form the viscous zone. Figure 3.1 shows how the ejector flow field is subdivided. The inviscid zone contains the ambient fluid that is drawn into the device. Inviscid flow exists inside a portion of the inlet between the jet and the channel wall. The viscous region originates at the jet nozzle and grows at a linear rate to simulate the spreading of the jet. The viscous zone completely fills the channel downstream of the point at which the jet first strikes the wall. The wake formed by the mixed flow which leaves the thrust augmentor exit is also part of the viscous zone, but it is ignored since calculations have shown [36] that it exerts a negligible effect on the mixing taking place within the channel.

The flow within the inviscid zone is also assumed to be irrotational and thus the solution can be generated under a potential flow framework. The flow is further assumed to be incompressible. A higher-order panel method is used as an efficient

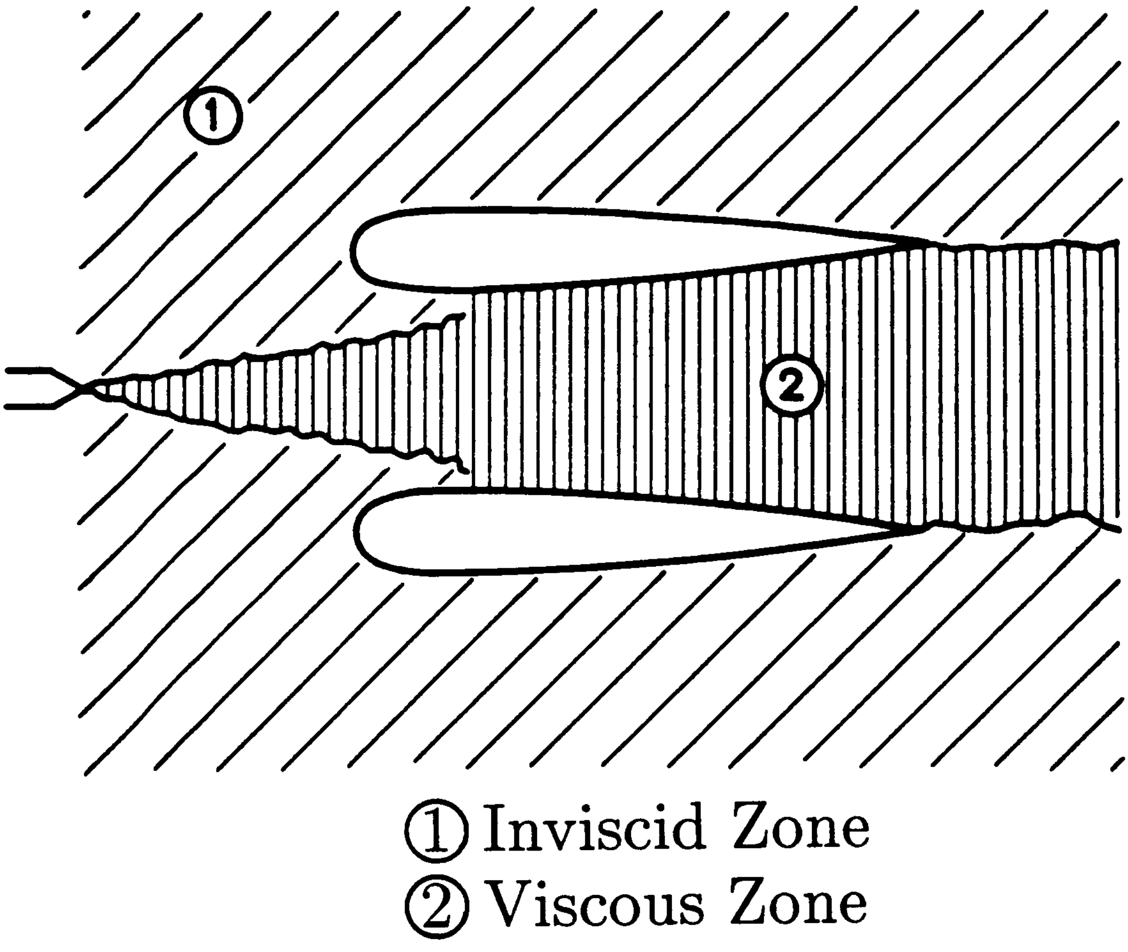


Figure 3.1: Subdivision of the ejector flow field into viscous and inviscid zones

means to produce accurate solutions for arbitrary shroud shapes. The effect of the jet entrainment on the inviscid field is simulated by applying suction to the panels which cover the jet boundary. The panel method is desirable since it does not require the use of a computational grid nor any iteration. The solution is formed by constructing and inverting a moderately sized matrix. The panel method has the added advantage that the panel suction boundary conditions only appear on the right hand side of the matrix equation. Thus, during the viscous-inviscid matching process, the matrix only needs to be calculated and inverted once. With each change in the jet entrainment distribution, the new inviscid solution is found through a simple matrix-vector multiply.

In the viscous region composed of the turbulent jet, streamwise diffusion is neglected, and thus the thin shear layer equations are used. These equations are solved in an integral formulation using the method of weighted residuals. In the integral formulation, the solution is efficiently obtained by assuming the form of the jet velocity profile. The velocity profile is made flexible by incorporating the secondary velocity, centerline velocity, and the jet growth rate as undetermined functions of the streamwise coordinate. The weighted residual procedure is applied to minimize the error introduced by the velocity profile assumption. This operation produces a set of first order differential equations for the functions that specify the velocity profile. The differential equations are integrated by marching downstream from the jet nozzle.

The viscous-inviscid procedure requires an iterative process to match the two zones together. To help understand the iteration process, the thrust augmentor is divided into two regions as shown in figure 3.2. In region 1, the jet merges with the co-flowing inviscid flow. This area is referred to as the interaction region since the viscous and inviscid flows are influencing each other here. Within this region the two solutions are matched together by iterating between the jet entrainment and the inviscid pressure distribution. In region 2, the turbulent zone completely fills the channel. In this region, the viscous flow is no longer influenced by the inviscid flow and no matching is needed.

The inviscid secondary flow present in the interaction region is produced by

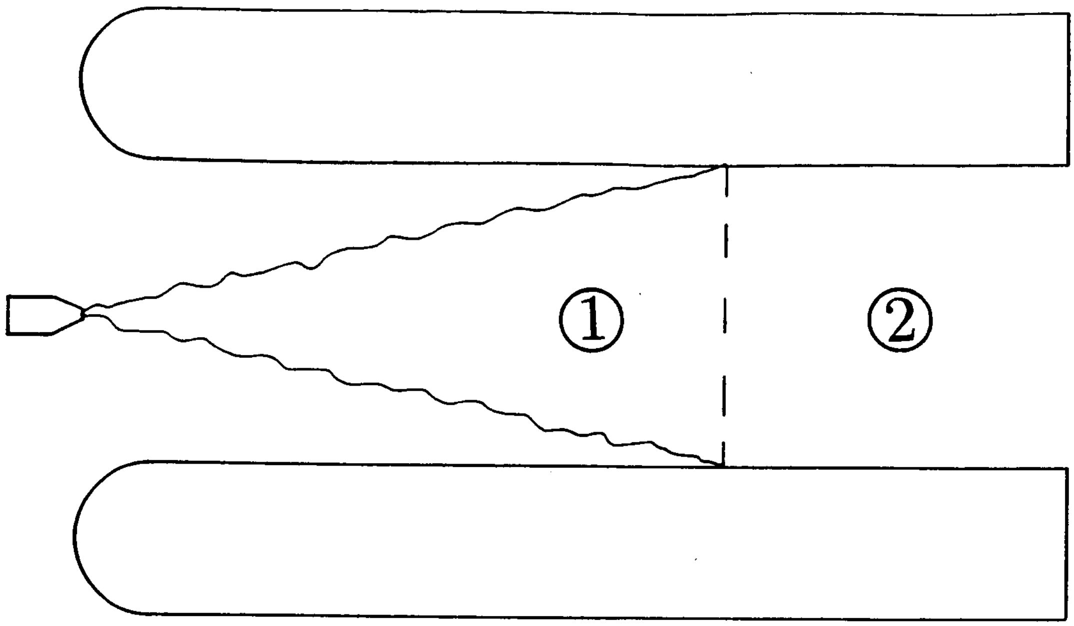


Figure 3.2: Viscous-inviscid interaction region. The viscous and inviscid flows are matched in region 1. In region 2 only the viscous equations are solved.

the jet entrainment. The secondary flow, in turn, influences the growth of the jet by imposing a pressure gradient and by reducing the rate of shear where the jet meets the ambient fluid. This coupling between the jet and secondary flow is simulated during the matching process. The inviscid solution provides the pressure gradient needed as a boundary condition to compute the viscous flow. The viscous solution is then used to produce a new distribution of jet entrainment. The panel suction velocities are updated and an improved inviscid solution is calculated. This procedure is continued until changes to the panel suction velocities are negligible.

Chapter 4

Inviscid Solution

4.1 Equations of Motion

The inviscid flow is assumed to be irrotational. The kinematics of the flow are then such that the velocity field may be described as the gradient of a scalar potential

$$\vec{U} = \nabla\Phi \tag{4.1}$$

If the above expression is substituted into the incompressible continuity relation

$$\nabla \cdot U = 0 \tag{4.2}$$

it is found that the velocity potential satisfies Laplace's equation

$$\nabla^2\Phi = 0 \tag{4.3}$$

An integral of the momentum equation for constant density gives the Bernoulli equation, which relates the pressure to the velocity field

$$p + 1/2\rho U^2 = const \tag{4.4}$$

Since Laplace's equation (and boundary conditions) are linear, solutions may be superimposed. Making use of this fact, the velocity potential is split into two parts; one corresponding to the free stream and another corresponding to the disturbance created by the body. Accordingly, The velocity potential is written as

$$\Phi = \phi_\infty + \phi \tag{4.5}$$

Under this formulation the velocity may be expressed as

$$\vec{U} = \vec{V}_\infty + \vec{V} \tag{4.6}$$

Since the potential due to the free stream is known, the problem involves finding the disturbance potential due to the presence of the body. The disturbance potential is not considered to be small as in thin airfoil theory. In the present work the potential is split into two parts for convenience, not for the purpose of linearization.

4.2 Solution Alternatives

While several methods are available for solving Laplace's equation, three of these should be given special consideration for the ejector problem. In particular, conformal mapping, finite differences, and panel methods are all means of producing accurate solutions in reasonable amounts of computational time. The individual merits and shortcomings of each are discussed below.

4.2.1 Conformal Mapping

The technique of conformal mapping[37] is probably the most efficient means of solving Laplace's equation. A conformal transformation is used to map the physical geometry into a simplified shape for which the solution of Laplace's equation is known. The solution to the physical problem is then found by applying the reverse transformation to the solution in the auxiliary plane. If the physical geometry is relatively simple, the transformation, and therefore the solution to Laplace's equation, can be determined analytically. For more complex geometries the transformation can not be determined analytically. In this case the conformal technique fails even if the transformation is determined numerically, since there is no direct way to determine the reverse transformation.

Tavella [13] used conformal mapping in his ejector model where the shroud was idealized as an infinitely thin flat plate. The geometry was then such that the classical Borda's mouthpiece solution [38] could be used. Tavella also was able to obtain solutions for shrouds that could be described in terms of small perturbations to the flat plates.

For the purposes of this work it is necessary to consider a more general class of shroud shapes. Since the conformal mapping technique is not applicable to an arbitrary geometry, it can not be used as the solution procedure in the present work.

4.2.2 Finite Difference Calculations

Finite difference methods[39] can solve a general class of inviscid flow problems. As long as a computational grid can be generated, the finite difference procedure will work on practically any geometry. Aside from the ability to handle general geometries, the finite difference method is unattractive in that a computational grid must be generated and that the solution requires a time-consuming iterative process. These features make finite difference methods computationally expensive, and therefore less attractive than the third alternative, panel methods.

4.2.3 Panel Methods

Panel methods[40] compete directly with finite difference methods in their ability to treat complex geometries. They are computationally cheaper than finite difference methods, however, since they do not require a computational grid or an iterative solution. The solution procedure involves solving a linear system of algebraic equations in a direct mode.

Of these three solution procedures, the panel method is the one preferred for the ejector study, since it is the most efficient method for the degree of generality required. Other methods such as the vortex lattice [41, chap. 7] or vortex sheet[42, chapt. 5] would work equally as well, but offer no further advantage over the panel method. The panel method is chosen since its use is well documented in the literature.

4.3 Derivation of the Source-Panel Method

Panel methods belong to a general class of surface singularity methods in which a solid body is replaced by distributions of various forms of singular elementary solutions to Laplace's equation (i.e. sources, doublets, vorticies, etc.) In the source-panel method used here, the body surface is broken up into a number of small elements or "panels" over which sources are distributed. The source intensities are determined by enforcing boundary conditions at the center of each element. For solid surfaces, the boundary condition is that the sum of the velocities induced by all of the panels exactly cancel the component of the free stream normal to the surface element. For flow-through boundaries, the sum of all the induced velocities and the free stream are required to equal a specified normal velocity.

4.3.1 Green's Third Identity

The starting point in the derivation of the panel method is the two-dimensional version of Green's third identity[43, page 142]

$$\phi(x_0, y_0) = \frac{1}{2\pi} \int_C \left[\frac{\partial \phi}{\partial n} \ln(r) - \phi \frac{\partial}{\partial n} \ln(r) \right] ds \tag{4.7}$$

Green's identity relates the value of the potential at any fixed field point (x_0, y_0) to an integral over the body contour. The distance from the fixed point (x_0, y_0) to the point of integration on the body surface is r, while n is the local outward pointing normal. $\frac{1}{2\pi}\ln(r)$ is the Green's function for a two-dimensional source. Its normal derivative, $\frac{1}{2\pi}\frac{\partial}{\partial n}\ln(r)$ represents a two-dimensional doublet. The derivative of potential normal to the surface has the interpretation of the source strength, while the value of the potential on the surface is associated with the doublet strength. These strengths are usually denoted by $\sigma(s)$ and $\mu(s)$ respectively. With these conventions, Eq. (4.7) becomes

$$\phi(x_0, y_0) = \frac{1}{2\pi} \int_C \left[\sigma(s) \ln(r) - \mu(s) \frac{\partial}{\partial n} \ln(r) \right] ds \tag{4.8}$$

Solutions to the boundary integral problem are not unique. That is several different distributions of σ and μ can be found to satisfy the given boundary conditions. In many cases a well conditioned problem can be formulated with the sources alone. The most notable of these is the flow over non-lifting aerodynamic bodies. It is easy to demonstrate that sources alone are also sufficient to produce aerodynamic forces, provided that the body is semi-infinite. In this work the ejector shroud is treated as a semi-infinite body and thus the solution can be obtained without the use of doublets.

In anticipation of discretizing the body surface into a collection of small elements, the boundary integral in Green's third identity is broken up into the sum of integrals taken over adjoining sections of the surface. Thus, using sources alone, Eq. (4.8) may be written equivalently as

$$\phi(x_0, y_0) = \sum_{j=1}^{N} \frac{1}{2\pi} \int_{s_j-\Delta s_j/2}^{s_j+\Delta s_j/2} \sigma(s) \ln r(s; x_0, y_0) ds \tag{4.9}$$

or more compactly

$$\phi(x_0, y_0) = \sum_{j=1}^{N} \frac{1}{2\pi} \int_{-\Delta s_j/2}^{\Delta s_j/2} \sigma(\hat{s}) \ln r(\hat{s}; x_0, y_0) d\hat{s} \tag{4.10}$$

where

$$\hat{s} = s - s_j \tag{4.11}$$

The derivation becomes simpler if each of the integrals contained in the above sum is transformed from its current curvilinear system to a local cartesian coordinate system placed tangent to the curve at $\hat{s} = 0$ (see Figure 4.1). The origin of the j^{th} local coordinate system lies at the point (x_{cp_j}, y_{cp_j}) in the global system. The subscript *cp* is used to denote control point since, later in the analysis, boundary conditions will be imposed at these points. The j^{th} transformation has the form

$$\begin{aligned} \xi_j &= (x - x_{cp_j}) \cos\alpha_j + (y - y_{cp_j}) \sin\alpha_j \\ \eta_j &= -(x - x_{cp_j}) \sin\alpha_j + (y - y_{cp_j}) \cos\alpha_j \end{aligned} \tag{4.12}$$

When the above transformation is used, Eq. (4.10) becomes

$$\phi(x_0, y_0) = \sum_{j=1}^{N} \frac{1}{2\pi} \int_{-\Delta\xi_j/2}^{\Delta\xi_j/2} \sigma(\xi) \ln r(\xi; \xi_{j0}, \eta_{j0}) \frac{d\hat{s}}{d\xi} d\xi \tag{4.13}$$

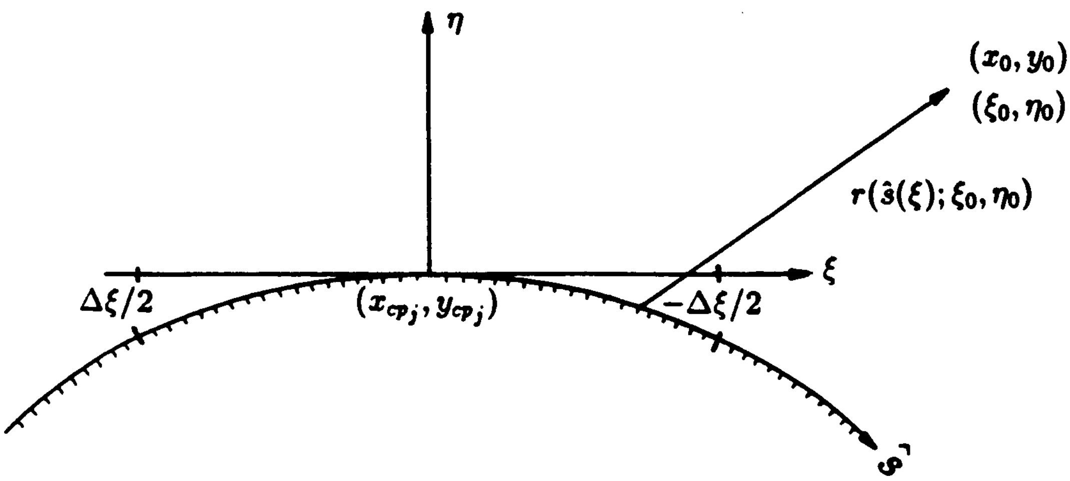


Figure 4.1: Higher order panel surface element

The distance from the transformed location of the fixed point (ξ_{j0}, η_{j0}) to the integration point on the body surface, $(\xi, \eta_b(\xi))$, is

$$r^2(\xi; \xi_{j0}, \eta_{j0}) = (\xi_{j0} - \xi)^2 + [\eta_{j0} - \eta_b(\xi)]^2 \tag{4.14}$$

The velocity components are found by differentiating Eq. (4.13) while making use of the chain rule and Eqs. (4.12) and (4.14) (the tildes are used to signify a velocity computed in the transformed coordinate system)

$$V_x(x_0, y_0) = \sum_{j=1}^{N} \left[\tilde{V}_{\xi_j}(\xi_{j0}, \eta_{j0}) \cos\alpha_j - \tilde{V}_{\eta_j}(\xi_{j0}, \eta_{j0}) \sin\alpha_j\right] \tag{4.15}$$

$$V_y(x_0, y_0) = \sum_{j=1}^{N} \left[-\tilde{V}_{\xi_j}(\xi_{j0}, \eta_{j0}) \sin\alpha_j + \tilde{V}_{\eta_j}(\xi_{j0}, \eta_{j0}) \cos\alpha_j\right] \tag{4.16}$$

where

$$\tilde{V}_{\xi_j}(\xi_{j0}, \eta_{j0}) = \frac{1}{2\pi} \int_{-\Delta\xi_j/2}^{\Delta\xi_j/2} \sigma(\xi) \frac{\xi_{j0} - \xi}{r^2} \frac{d\hat{s}}{d\xi} d\xi \tag{4.17}$$

$$\tilde{V}_{\eta_j}(\xi_{j0}, \eta_{j0}) = \frac{1}{2\pi} \int_{-\Delta\xi_j/2}^{\Delta\xi_j/2} \sigma(\xi) \frac{\eta_{j0} - \eta_b(\xi)}{r^2} \frac{d\hat{s}}{d\xi} d\xi \tag{4.18}$$

4.3.2 Taylor Series Expansion of the Source Intensity and Surface Shape

Up to this point no approximation has been introduced. If the distribution of source intensity $\sigma(\xi)$ and the body shape $\eta_b(\xi)$ are known, then the above relations can be used to find the exact result for velocity at any point in the field. This is not possible in general, however, because the source distribution is not known a priori. In the panel method, both the source strength and the surface shape are expanded Taylor series centered about the element origin. If the element length is small compared with the distance to the field point, higher order terms in the expansion are small and can be neglected.

4.3.3 Classical Panel Method

In the classical or "first order" panel method, only the leading terms in each of the expansions are retained. That is, the body shape is approximated by a collection of linear segments and the source strength is taken to be locally constant over each of the elements. Under this approximation, the surface description contains slope discontinuities, while the source distribution is discontinuous in strength. The discontinuity in source strength that occurs at the panel junctions has a repercussion in the solution in that the velocity becomes infinite at these points. The velocity remains well behaved at the panel center, however, and the method can be used to produce accurate results provided that the surface velocity calculations are restricted to the panel center.

An additional problem associated with the discontinuous source strength is that the body "leaks" mass at the panel junctions. Due to to a fortuitous symmetric cancellation of errors, the leaks do not pose a serious difficulty when external flows are computed.[40]. However, the situation is reversed when computing internal flows since error reinforcement spoils the solution[44]. If the classical panel method is applied to a duct flow problem, mass will not be conserved within the duct. In addition, the velocity field becomes singular near bends in the channel wall or at the duct end. Attempts to remedy this problem by decreasing the panel size are not met with success since the first order approximation converges slowly to the exact solution. A prohibitively large number of panels would be needed to accurately describe an internal flow with the classical panel method.

4.3.4 Higher Order Panel Method

The best way to avoid the leakage problem in internal flows is to retain more terms in the expansions for the source distribution and surface shape. In the higher order method described by Hess[45], the body is described by quadratic surface elements. The singularity intensity is also allowed to vary quadratically. This formulation makes both the surface shape and source distribution continuous through the first derivative. The leakage problem is eliminated and the approximate solution converges to the exact one with the third power of the ratio of the panel length to the distance to the field point. The formulation of Hess is adopted here and the derivation which follows is similar to the less detailed account given in the original paper[45].

The derivation is started by introducing higher order approximations to the source distribution and the body surface shape. To second order, these quantities may be approximated as

$$\sigma(\xi) = \sigma_j + \dot{\sigma}_j\xi + \frac{1}{2}\ddot{\sigma}_j\xi^2 + O(\xi^3) \tag{4.19}$$

$$\begin{aligned} \eta_b(\xi) &= \tfrac{1}{2}\tfrac{\partial^2\eta}{\partial\xi^2}\xi^2 + O(\xi^3) \\ &= \tfrac{1}{2}\kappa_j\xi^2 + O(\xi^3) \end{aligned} \tag{4.20}$$

where κ_j is the local surface curvature. Note that, because the origin of the local coordinate system is tangent to the body curve, the first two terms of the Taylor series for the surface shape are zero. The arc length along the surface is found from

$$\hat{s} = \int_0^\xi \sqrt{1 + \left(\frac{d\eta_b}{d\xi}\right)^2}\, d\xi \tag{4.21}$$

Eq. (4.20) is substituted above, and the resulting integral expanded for small ξ to give

$$\hat{s} = \xi + \frac{1}{6}\kappa_j^2\xi^3 + O(\xi^4) \tag{4.22}$$

The jacobian for the change of variables between $\hat{s}$ and ξ is

$$\frac{d\hat{s}}{d\xi} = 1 + \frac{1}{2}\kappa_j^2\xi^2 + O(\xi^3) \tag{4.23}$$

Using Eqs. (4.14), (4.19), (4.20), and (4.23), the integrands of the expressions for the velocity components (Eqs. (4.17) and (4.18)) can be expressed as functions of ξ and then expanded for small ξ. The resulting integrals give the velocity components as a power series in $\Delta\xi$. This series will not converge, however, if the velocity is computed at a point closer than $\Delta\xi/2$ from the panel center. This problem is relieved by using an alternate expansion of the integrands in which terms that correspond to a constant source strength over a flat panel are retained as functions of ξ, and all other terms expanded for small ξ. The resulting integrals involve the same terms that are found in a classical panel method, plus terms proportional to powers of $\Delta\xi$ that represent the contributions from the higher order effects of surface and singularity shape. This formulation not only makes the series convergent, but at the same time, assures that the integrals reduce to the results for a classical panel method as the surface element becomes vanishingly small.

The modified expansion is implemented by first writing Eq. (4.14) in the following equivalent form

$$\begin{aligned} r^2 &= \left[(\xi_{jo} - \xi)^2 + \eta_{jo}^2\right] - 2\eta_{jo}\eta_b(\xi) + \eta_{jo}^2 \\ &= r_f^2 - 2\eta_{jo}\eta_b(\xi) + \eta_{jo}^2 \end{aligned} \tag{4.24}$$

The quantity r_f represents the distance from the field point to a point on a "flat" element sitting on the ξ axis (see Figure 4.1). Equation (4.20) is now used to write the above expression as

$$r^2 = r_f^2\left[1 - \kappa\left(\frac{\eta_{jo}\xi}{r_f^2}\right)\xi + \frac{1}{4}\eta_{jo}\kappa^2\left(\frac{\eta_{jo}\xi}{r_f^2}\right)\xi^3\right] \tag{4.25}$$

Note that when r_f is small, r_f, η_{jo}, and ξ are all of the same order of magnitude. Thus the grouping $\eta_{jo}\xi/r_f^2$ remains of order unity as r_f drops below $\Delta\xi/2$. The series for the induced velocities will converge if the quantity r_f is retained as a function of ξ in the integration, and only the latter terms of Eq. (4.25) expanded.

In particular

$$\frac{1}{r^2} = \frac{1}{r_j^2}\left[1 + \kappa\left(\frac{\eta_{jo}\xi}{r_j^2}\right)\xi + O(\xi^3)\right] \tag{4.26}$$

The above expression as well as Eqs. (4.20), (4.23), and (4.19) are substituted into the integrals of Eqs. (4.17), and (4.18). The resulting expressions are expanded in powers of ξ, and terms through order ξ^2 retained to give the velocity components induced by the j^{th} panel

$$\begin{aligned} \tilde{V}_{\xi_j}(\xi_{jo}, \eta_{jo}) &= \frac{1}{2\pi}\int_{-\Delta\xi_j/2}^{\Delta\xi_j/2} \frac{(\xi_{jo} - \xi)}{r_j^2}\left\{\sigma_j + \left[\kappa_j\left(\frac{\eta_{jo}\xi}{r_j^2}\right)\sigma_j + \dot{\sigma}_j\right]\xi + \right. \\ &\left. \frac{1}{2}\left(\kappa_j^2\sigma_j + \ddot{\sigma}_j\right)\xi^2 + O(\xi^3)\right\} d\xi \end{aligned} \tag{4.27}$$

$$\begin{aligned} \tilde{V}_{\eta_j}(\xi_{jo}, \eta_{jo}) &= \frac{1}{2\pi}\int_{-\Delta\xi_j/2}^{\Delta\xi_j/2} \frac{\eta_{jo}}{r_j^2}\left\{\sigma_j + \left[\kappa_j\left(\frac{\eta_{jo}\xi}{r_j^2} + \frac{1}{2}\frac{\xi}{\eta_{jo}}\right)\sigma_j + \dot{\sigma}_j\right]\xi + \right. \\ &\left. \frac{1}{2}\left(\kappa_j^2\sigma_j + \ddot{\sigma}_j\right)\xi^2 + O(\xi^3)\right\} d\xi \end{aligned} \tag{4.28}$$

When the above integrals are evaluated, the results may be written in vector form as

$$\vec{\tilde{V}}_j(\xi_{jo}, \eta_{jo}) = \vec{\tilde{A}}_j^{(0)}\sigma_j + \left[\vec{\tilde{A}}_j^{(c)}\kappa_j\sigma_j + \vec{\tilde{A}}_j^{(1)}\dot{\sigma}_j\right]\Delta\xi_j + \vec{\tilde{A}}_j^{(2)}\left[\kappa_j^2\sigma_j + \ddot{\sigma}_j\right]\Delta\xi_j^2 \tag{4.29}$$

Here $\vec{\tilde{A}}^{(0)}$ represents the disturbance due to constant source strength distributed over a flat surface element. This is the only term which is resolved in a classical panel method. The next higher order term is composed of two parts, one that accounts for the surface curvature and another that accounts for the slope of the source intensity. The last term above involves still higher order effects of surface and singularity distribution curvature. The individual terms in Eq. (4.29) are written out in full below. With the definitions

$$\begin{aligned} r_1^2 &= (\xi_0 + \Delta\xi/2)^2 + \eta_0^2 \\ r_2^2 &= (\xi_0 - \Delta\xi/2)^2 + \eta_0^2 \end{aligned} \tag{4.30}$$

the individual terms in Eq. (4.29) are

$$
\begin{aligned}
\tilde{A}_\xi^{(0)} &= \tfrac{1}{4\pi}\ln\tfrac{r_1^2}{r_2^2} \\
\tilde{A}_\eta^{(0)} &= \tfrac{1}{2\pi}\left[\tan^{-1}\left(\tfrac{\xi_0+\Delta\xi/2}{\eta_0}\right) - \tan^{-1}\left(\tfrac{\xi_0-\Delta\xi/2}{\eta_0}\right)\right] \\
\tilde{A}_\xi^{(1)} &= \tfrac{1}{\Delta\xi}\left[\eta_0\tilde{A}_\eta^{(0)} + \xi_0\tilde{A}_\xi^{(0)} - 2\Delta\xi\right] \\
\tilde{A}_\eta^{(1)} &= \tfrac{1}{\Delta\xi}\left[\xi_0\tilde{A}_\eta^{(0)} - \eta_0\tilde{A}_\xi^{(0)}\right] \\
\tilde{A}_\xi^{(c)} &= \tfrac{1}{\Delta\xi}\left[-\xi_0\tilde{A}_\eta^{(0)} + \eta_0 v_\xi^{(0)} + \tfrac{1}{2}\tfrac{\xi_0\eta_0\Delta\xi^3}{r_1^2 r_2^2}\right] \\
\tilde{A}_\eta^{(c)} &= \tfrac{1}{\Delta\xi}\left[\eta_0\tilde{A}_\eta^{(0)} + \xi_0\tilde{A}_\xi^{(0)} - \Delta\xi\left(1+\tfrac{(\xi_0^2+\eta_0^2)^2-(\xi_0^2-\eta_0^2)(\Delta\xi/2)^2}{r_1^2 r_2^2}\right)\right] \\
\tilde{A}_\xi^{(2)} &= \tfrac{1}{\Delta\xi^2}\left[\xi_0\eta_0\tilde{A}_\eta^{(0)} + \tfrac{1}{2}(\xi_0^2-\eta_0^2)\tilde{A}_\xi^{(0)} - \xi_0\Delta\xi\right] \\
\tilde{A}_\eta^{(2)} &= \tfrac{1}{\Delta\xi^2}\left[\tfrac{1}{2}(\xi_0^2-\eta_0^2)\tilde{A}_\eta^{(0)} - \eta_0\xi_0\tilde{A}_\xi^{(0)} + \eta_0\Delta\xi\right]
\end{aligned}
\tag{4.31}
$$

These formulas give the velocity induced by the j^{th} panel in terms of its local coordinate system. It is more useful to have the velocity in terms of the global coordinate system. Using Eqs. (4.15) and (4.16), it is possible to write each of the above terms in the following general form

$$
\begin{aligned}
A_{x_j} &= \tilde{A}_{\xi_j}\cos\alpha_j - \tilde{A}_{\eta_j}\sin\alpha_j \\
A_{y_j} &= \tilde{A}_{\xi_j}\sin\alpha_j + \tilde{A}_{\eta_j}\cos\alpha_j
\end{aligned}
\tag{4.32}
$$

The derivatives of the source distribution still remain to be determined. This is done by using second order accurate finite differences as follows

$$
\begin{aligned}
\dot{\sigma}_j &= D_j\sigma_{j-1} + E_j\sigma_j + F_j\sigma_{j+1} \\
\ddot{\sigma}_j &= G_j\sigma_{j-1} + H_j\sigma_j + I_j\sigma_{j+1}
\end{aligned}
\tag{4.33}
$$

where

$$
\begin{aligned}
D_j &= -\tfrac{b}{a(a+b)} \\
E_j &= \tfrac{b-a}{ab} \\
F_j &= \tfrac{a}{b(a+b)} \\
G_j &= \tfrac{2}{a(a+b)} \\
H_j &= -\tfrac{2}{ab} \\
I_j &= \tfrac{2}{b(a+b)}
\end{aligned}
\tag{4.34}
$$

and where

$$\begin{aligned} a &= \tfrac{1}{2}(\Delta\xi_{j-1} + \Delta\xi_j) \\ b &= \tfrac{1}{2}(\Delta\xi_j + \Delta\xi_{j+1}) \end{aligned} \tag{4.35}$$

4.3.5 Boundary Conditions

In order to satisfy the boundary conditions, it is necessary to determine the net influence of all panels acting at the control point of the i^{th} panel. This is done by systematically finding the influence of the j^{th} panel at the fixed control point i, and then summing the results over all j. When considering the j^{th} panel, the point (ξ_{j_0}, η_{j_0}) in Eq. (4.29) is made to correspond to the i^{th} panel control point. Next Eq. (4.32) is used to transform the influence of the j^{th} panel into the global coordinate system. Equation (4.33) is then used to replace the derivatives of the source distribution in terms of the values at the panel center as well at the two adjacent panel centers. Finally the results are summed to give

$$\begin{aligned} \vec{V}_i &= \sum_{j=1}^{N} \Big\{ \vec{A}_{ij}^{(0)}\sigma_j + \left[\vec{A}_{ij}^{(c)}\kappa_j\sigma_j + \vec{A}_{ij}^{(1)}\left(D_j\sigma_{j-1} + E_j\sigma_j + F_j\sigma_{j+1}\right)\right]\Delta\xi_j + \\ &\quad \vec{A}_{ij}^{(2)}\left[\kappa_j^2\sigma_j + \left(G_j\sigma_{j-1} + H_j\sigma_j + I_j\sigma_{j+1}\right)\right]\Delta\xi_j^2 \Big\} \end{aligned} \tag{4.36}$$

The above sum may be written equivalently as

$$\vec{V}_i = \sum_{j=1}^{N} \vec{B}_{ij}\sigma_j \tag{4.37}$$

where

$$\begin{aligned} \vec{B}_{ij} &= \vec{A}_{ij}^{(0)} + \vec{A}_{ij}^{(1)}E_j\Delta\xi_j + \vec{A}_{ij}^{(c)}\kappa_j\Delta\xi_j + \vec{A}_{ij}^{(2)}(H_j + \kappa_j^2)\Delta\xi_j^2 + \\ &\quad \vec{A}_{ij-1}^{(1)}F_{j-1}\Delta\xi_{j-1} + \vec{A}_{ij-1}^{(2)}I_{j-1}\Delta\xi_j^2 + \\ &\quad \vec{A}_{ij+1}^{(1)}D_{j+1}\Delta\xi_{j+1} + \vec{A}_{ij+1}^{(2)}G_{j+1}\Delta\xi_j^2 \end{aligned} \tag{4.38}$$

The quantity $\vec{B}_{ij}$ is interpreted as the influence of the j^{th} panel at the i^{th} control point. A linear system of algebraic equations for the source strengths is formed by imposing one boundary condition per panel. The boundary condition is that the velocity normal to the panel have a specified value, that is

$$(\vec{V}_i + \vec{V}_\infty)\cdot\vec{n}_i = V_{n_i} \tag{4.39}$$

where $\vec{n}$ is the outward pointing normal defined as

$$\vec{n}_i = -\sin\alpha_i\hat{x} + \cos\alpha_i\hat{y} \tag{4.40}$$

For solid surfaces, V_n is zero. Non-zero values of V_n correspond to flow-through boundaries or porous surfaces. When each of the boundary conditions are enforced, the following linear system arises

$$\sum_{j=1}^{N}(-\sin\alpha_i B_{x_{ij}} + \cos\alpha_i B_{y_{ij}})\sigma_j = V_{x_\infty}\sin\alpha_i - V_{y_\infty}\cos\alpha_i + V_{n_i} \tag{4.41}$$

The solution of this matrix equation yields the source strength values σ_j and the velocity at each of the control points can be found through the use of Eq. (4.37). Velocities at any other arbitrary point in the field may be calculated by following the procedure used to generate Eq. (4.36), where the i^{th} control point is replaced by the field point.

4.3.6 Surface Curvature Calculation

One remaining detail of the higher-order panel method is a procedure for calculating the surface curvature. If the body surface is described by an analytic function, the curvature is known everywhere, and the procedure is straightforward. In most instances, however, the geometry is not described by an analytic expression, but rather by $N + 1$ discrete points on the body surface. In this case, the curvature must be computed by a suitable approximate means. A good way to do this is to use a pair of parametric spline fits[1] where x_b and y_b are treated as functions of the approximate arc length found by summing the linear distance between points. Let ζ be the approximate arc length, then the spline fits give

$$\begin{aligned} x_b &= x_b(\zeta) \\ y_b &= y_b(\zeta) \end{aligned} \tag{4.42}$$

[1]A pair of spline fits is needed since in general the surface can not be described by y_b as a single-valued function of x_b.

Let $\dot{x}_b$ and $\dot{y}_b$ denote derivatives with respect to ζ. Then the curvature for the j^{th} panel is computed from [46, page 464]

$$\kappa_j = \left.\frac{\dot{x}_b\ddot{y}_b - \dot{y}_b\ddot{x}_b}{(\dot{x}_b{}^2 + \dot{y}_b{}^2)^{3/2}}\right|_{\zeta=j+1/2} \tag{4.43}$$

4.4 Inviscid Solution for the Single-Jet Ejector

The panel method requires that the surface of the ejector be broken up into a collection of small surface elements. Since the configurations treated here are symmetric, it is sufficient to consider only the upper half plane. Figure 4.2 shows how the upper half plane of the single-jet ejector is modeled with the panel method. The dividing streamline that approaches the ejector along the plane of symmetry is treated as a solid boundary. The following sloped linear segment represents the boundary between the viscous jet flow and inviscid secondary flow. The angle between this segment and the jet axis is taken to be 12^0 in accord with observations for the spreading rate of free jets. The position of the panels that cover the jet boundary remain fixed during the calculation. If the jet spreads less than the assumed 12^0, some of the inviscid flow will be contained within the viscous region. This does not present a problem, however, since the viscous formulation is also able to handle the inviscid portion of the flow, provided that it is uniform. Suction boundary conditions are applied to the panels that cover the jet to simulate entrainment of the secondary flow. The magnitude of the suction applied at the jet boundary panels is determined in the solution process. The half-circle at the upper end of the jet boundary serves as a control station where a uniform flow boundary condition is applied. The need for the control station arises from the fact that panel methods become inaccurate inside the sharp concave corner that would otherwise exist where the jet intersects the ejector channel walls. The uniform flow boundary condition is justifiable since experiments have shown [47] that the secondary flow well within the channel becomes nearly uniform.

The ejector shroud is modeled as an impermeable surface. The wake formed behind the ejector is treated as a continuation of the same streamline that defines the shroud. Under this assumption the mixing taking place in the wake is neglected.

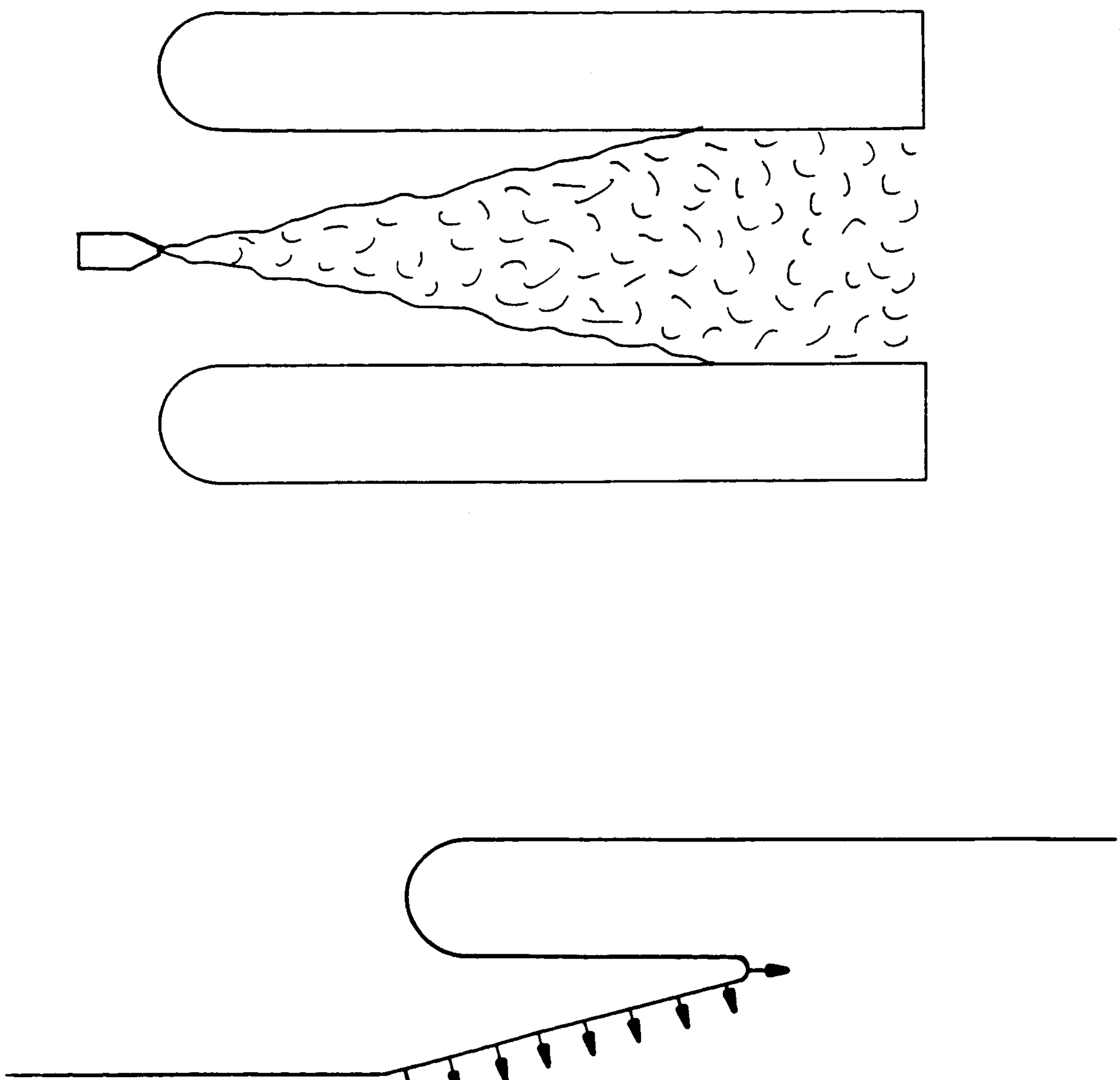


Figure 4.2: Panel geometry for the single-jet ejector

This is justifiable since computations have shown that the details of the wake have little effect on the performance of the ejector.

4.5 Inviscid Solution for the Dual-Jet Ejector

The panel geometry used for the dual-jet ejector model is quite similar to that used in the single-jet model. The actual distribution of panels is shown in Figure 4.3. As in the single-jet case, the presence of symmetry allows the solution to be restricted to the upper half plane. Unlike the single-jet ejector case, the upper half plane for the dual-jet ejector contains one whole jet. The entrainment that occurs on both the upper and lower side of the jet is accounted for by applying suction to the panels that cover both sides of the jet. To account for asymmetries in the secondary flow with respect to the jet centerline, The distribution of entrainment on either side of the jet is not required to be the same. The distribution of entrainment velocities for both sides of the jet are again determined in the solution process.

Owing to a non-uniform pressure profile in the secondary flow near the ejector inlet, the jet is acted upon by a transverse pressure difference. The jet responds to the pressure difference by curving its trajectory in such a way that the centrifugal force acting on the fluid particles is balanced by the force created by the pressure difference. The inviscid solution accounts for this by placing jet boundary panels on curved surfaces that reflect the curvature of the jet centerline. The shape of the curved jet trajectory is not known a priori, but rather must be determined along with the rest of the solution. For this reason, the panels that cover the jet boundary in the dual-jet case must be free to move as the solution progresses. After each iteration, the locations of the jet boundary panels are adjusted to conform with the latest computation of the jet trajectory.

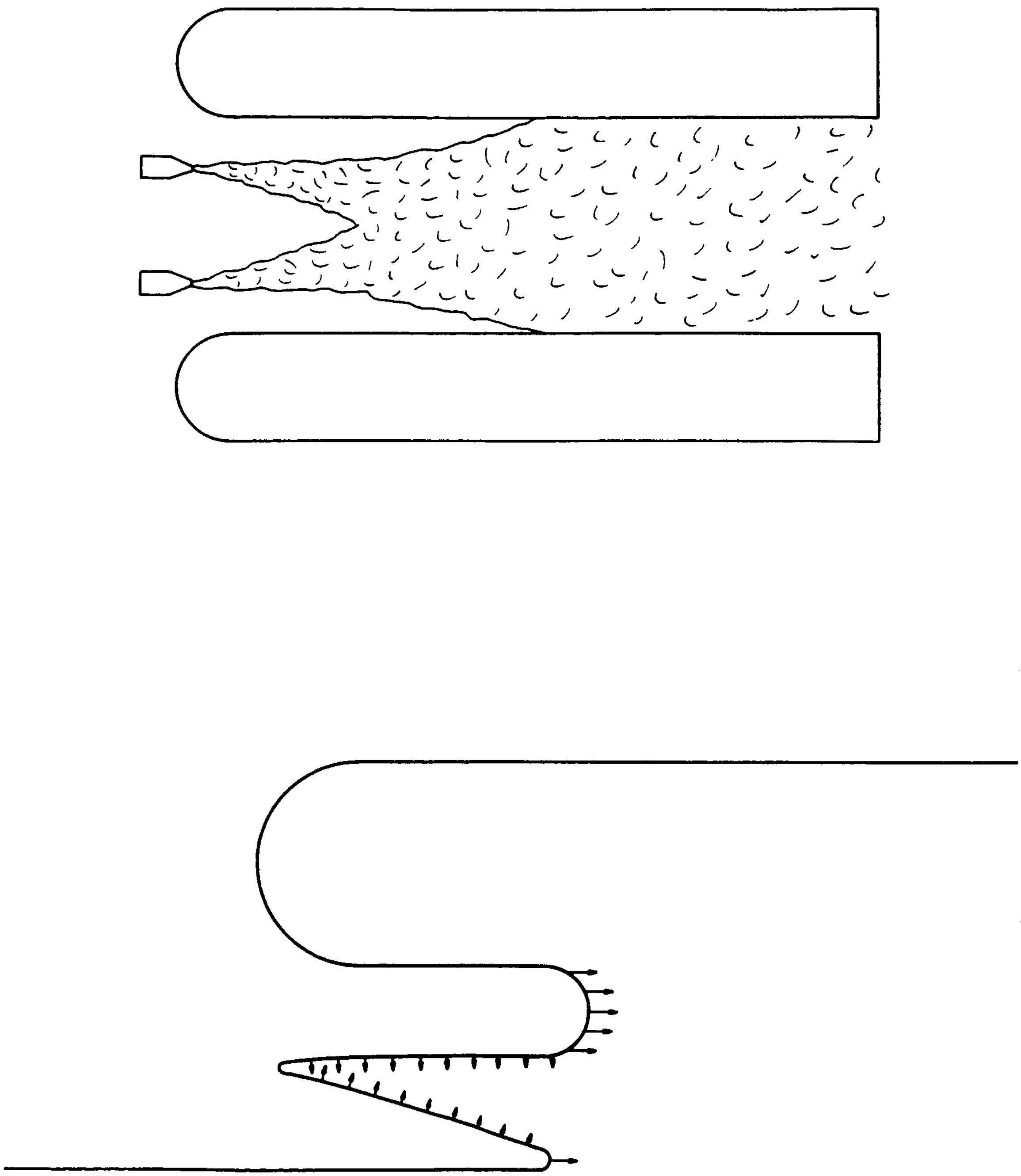


Figure 4.3: Panel geometry for the dual-jet ejector

Chapter 5

Viscous Solution

5.1 Equations of Motion

Turbulent jets are similar to boundary layers in that their transverse extent is small when compared with their streamwise length. The fluid shear is contained within a thin layer near the jet axis, and thus the streamwise gradients are small when compared with the transverse gradients. Under these conditions, the boundary layer assumptions are met and it is permissible to neglect the streamwise diffusion term in the Navier-Stokes equations. In addition, turbulent jets have the special characteristic that they develop in the absence of solid surfaces, where molecular viscosity is an important factor. Turbulent transport dominates molecular transport everywhere in the jet flow field, and it is therefore possible to entirely neglect the effects of viscosity[48, page 53].

The equations that govern the jet flow are the turbulent boundary layer equations in which the molecular viscosity has been neglected.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.1}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} = \frac{1}{\rho}\frac{\partial \tau}{\partial y} \tag{5.2}$$

The transverse momentum equation is retained in the following approximate form[49] that relates the lateral pressure gradient to the centrifugal force associated with curved particle trajectories

$$\frac{\partial p}{\partial y} = \frac{\rho u^2}{R} \tag{5.3}$$

The flow is assumed to be incompressible, and thus the equation of state is simply

$$\rho = \text{const} \tag{5.4}$$

Finally, the turbulent shear stress is related to the mean velocity gradients via the Boussinesq approximation

$$\frac{1}{\rho}\tau = \nu_t \frac{\partial u}{\partial y} \tag{5.5}$$

Here ν_t is the "eddy viscosity coefficient" which is determined from a simple algebraic stress model.

5.2 Solution Alternatives

The boundary layer equations are classified mathematically as being parabolic. Parabolic equations are relatively simple to solve since the properties at any given station are only affected by the upstream flow history. This one-sidedness allows approximate solution methods to be formulated in terms of simple marching schemes that integrate the equations in a single streamwise pass. The merits of a few suitable approximate schemes, as well an exact solution alternative are considered below.

5.2.1 Similarity Solutions

In a few special cases, the boundary conditions are such that the boundary layer equations yield exact solutions. These solutions are all of the similarity type, in which the absence of a natural length scale dictates that the solution must depend on the ratio y/x. This regrouping of variables reduces the dimension of the problem by one, and the boundary layer equations reduce to an integrable ordinary differential equation.

The turbulent free jet is one such special case. A similarity solution to the free turbulent free jet was first found by Tollmien[50] in 1926. Tollmien, who used Prandtl's mixing length formula, arrived at the solution in terms of a modified stream function that had to be found numerically. Later Gortler[51] used an eddy viscosity model to arrive at a purely analytical result in which the solution is written in terms of hyperbolic functions. These solutions are extremely valuable since they give the velocity everywhere in the field in terms of a single known function.

While similarity solutions exist for the free jet, they do not, in general, exist for confined jets. The separation between the channel walls, the external flow velocity, and the pressure gradient all introduce length scales that spoil similarity. Newmann[52] performed a detailed study of the conditions under which self-similar solutions exist for jets subjected to a streamwise pressure gradient. He found that similarity is only possible under the following restrictive conditions on the external flow

$$\frac{u_0}{u_1} = \text{const} \tag{5.6}$$

and

$$\frac{\dot{u}_0 b}{u_0} = \text{const} \tag{5.7}$$

where u_0 is the external velocity, u_1 the jet excess velocity ($u_{max} - u_0$), and b the excess velocity half-width. These are strong conditions which are not expected to be satisfied in ejector flows.

While similarity solutions do not in general exist for ejector flows, some previous investigators[9,53,54] have nonetheless incorporated the self-similar solution to a free jet in their analysis. In these works, it was assumed that the free jet solution would provide a reasonable estimate of the mixing process within the ejector. Experiments[47] do not support this assumption, however, and in fact show a significant departure from self-similarity with downstream distance. The use of free jet solutions may be acceptable in low accuracy solutions of relatively short ejectors, but should be ruled out in work aimed at a better understanding of the ejector mixing process.

5.2.2 Finite Difference Methods

A more general method of solving the confined jet problem is through a finite difference calculation of the boundary layer equations. Unlike the elliptic equations encountered in the inviscid flow, the boundary layer equations may be solved under a finite difference scheme that does not involve iteration[55, chapt. 7] and [56]. A grid must still be generated, but the solution is obtained in a straightforward marching process, in which a tri-diagonal system of equations is inverted at each streamwise location. Finite difference procedures for solving the viscous flow within the ejector have been used in the past by several investigators [11,57,58].

Although the finite difference approach is relatively efficient, it still is not fast enough for the purposes of this work. Fortunately, an alternative method, superior in efficiency, exists for solving confined jet flows. This is the "momentum integral method" or more simply "integral method".

5.2.3 Integral Methods

An integral method is a form of approximation that does not attempt to satisfy the boundary layer equations at every point, but rather only satisfies the equations on an average sense over the thickness of the shear layer. This is accomplished by first integrating the boundary layer equations across the layer, and then finding an approximate solution to the resulting integro-differential equation. This approximate solution is found by assuming that the velocity profile has the same shape at each streamwise location, and that it is only the relative scaling of the profile which changes as the flow evolves. This idea allows the velocity profile to be expanded in terms of assumed shape functions of y, but undetermined scale functions of x. When the approximate expression is substituted into the integral form of the momentum equation, the integral in y can be performed analytically. What remains is a coupled set of ordinary differential equations for the scale functions of x. Only a trivial amount of computing effort is needed to march the solution of this set equations in the streamwise direction.

The integral method was first applied to boundary layer flows by von Karman [59] and later refined by Pohlhausen[60]. In these original works the velocity profile was expanded in a fourth order polynomial of $y/\delta(x)$, where $\delta(x)$ is the boundary layer thickness. The problem was thus reduced to solving a single ordinary differential equation for the scale function $\delta(x)$. This solution procedure is extremely efficient and surprisingly accurate; the Karman-Pohlhausen solution predicts the skin friction to within 3.5% of the exact solution for a flat plate boundary layer.

Integral methods have also been successfully applied to confined jet flows. Curtet[61] developed an integral method for confined jets which was valid over the region where a definable inviscid flow co-exists with the jet in the channel. Hill[62] extended this analysis to the region where the flow within the channel is fully turbulent. Bevilaqua [9] and Tavella[12] have refined the method still further, and have applied it to ejector flows. Tavella compared his results with experiments and found a good agreement for the velocity profile and pressure evolution. Tavella's solution proved not only to be accurate, but extremely efficient as well. The four differential equations in his model could be marched through the ejector in a fraction of a second on an IBM 30-81 processor. In light of the previously demonstrated accuracy and economy of the integral method, it is adopted here as the preferred solution procedure for the viscous region.

The velocity profiles which are used in this work involve several scale functions of x. In this case, the integrated momentum equation itself does not provide enough information to determine one differential equation for each of the scale functions. The system is closed by using the method of weighted residuals to generate additional differential equations for the scale functions.

5.3 The Method of Weighted Residuals

The method of weighted residuals is a particular solution procedure for the integral method that allows an arbitrary number of independent differential equations for the scale functions to be generated from the momentum equation. A special application of the method is developed to produce a square system of equations for the scale functions by simultaneously enforcing an exact global conservation of mass and momentum, while enforcing an approximate global conservation of energy. This formulation leads to a condition that requires the residual error, created when the approximate velocity and pressure profiles are substituted into the momentum equation, be minimized. Minimization is achieved by demanding that the error be orthogonal to an independent set of weighting functions.

The derivation of the method is straightforward. The first step is to integrate the continuity equation Eq. (5.1) with respect to y to give v as a function of u

$$v = -\int_{y_1}^{y} \frac{\partial u}{\partial x} dy \tag{5.8}$$

The lower limit in the integration is jet centerline, y_1, where the v component of velocity vanishes by symmetry. Next let Γ be the operation on u and p that represents the streamwise momentum equation. Then Eqs. (5.2), (5.5), and (5.8) may be combined to give

$$\Gamma\{u, p\} = u\frac{\partial u}{\partial x} - \left[\int_{y_1}^{y} \frac{\partial u}{\partial x} dy\right] \frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} - \nu_t \frac{\partial^2 u}{\partial y^2} = 0 \tag{5.9}$$

Approximate solutions for the velocity and pressure profile are now introduced. The assumed profiles depend explicitly on y through the known shape functions, and implicitly on x through the unknown scale functions. Let the scale functions be denoted by the sequence $c_j(x)$, then the approximate solution forms (denoted by hats) may be written symbolically as

$$u(x, y) \simeq \hat{u}(c_j(x), y) \qquad j = 1, 2, ..., N-1 \tag{5.10}$$

$$\begin{aligned} p(x, y) &\simeq \hat{p}(c_j(x), y) && j = 1, 2, ..., N \\ &= c_N(x) + \tilde{p}(c_j(x), y) && j = 1, 2, 3, ...N-1 \end{aligned} \tag{5.11}$$

Note that the pressure has been split in two parts; one a function of x alone, and the other a function of both x and y. The elliptic effects associated with the pressure field are captured by taking $\tilde{p}$ to be the solution of the approximate transverse momentum equation (Eq. (5.3)). The function $\tilde{p}$ is an order of magnitude lower than c_N, and therefore is a higher order term in the streamwise momentum equation. In the usual procedure for the partially parabolized Navier-Stokes equations[55], $\tilde{p}$ is neglected in the streamwise momentum equation and retained only in the transverse momentum equation. In this work $\tilde{p}$ could also justifiably be neglected in the streamwise momentum equation. This is not done, however, since if $\tilde{p}$ were neglected, Bernoulli's equation for the inviscid flow would not be exactly recovered at far distances from the jet centerline.

In order to obtain a set of equations to determine the unknown scale functions, the momentum equation is transformed from a statement of local flux balances to one of global flux balance by integrating it across the layer. If the viscous region extends from $y = 0$ to $y = H$, the equation for the global conservation of momentum applied to the approximate profiles $\hat{u}$ and $\hat{p}$ provides the following governing equation for these quantities

$$\int_0^H \Gamma(\hat{u}, \hat{p}) dy = 0 \tag{5.12}$$

Use of the above averaged form of the momentum equation to specify the approximate profile leads to weaker solutions than those for the original differential form of the momentum equation. Although exact solutions to the above integral equation can easily be found, they will not satisfy the differential form of the momentum equation at each point. The weighted residual method provides a means of minimizing the error, however, and the weak solutions may used as a good approximation. In fact, when a properly implemented weighted residual method is used, the approximate solution will rapidly converge to the exact solution as the assumed profiles become increasingly flexible.

When the integral formulation is used, a subtle point arises in connection with the global conservation of mechanical energy. The equation that governs the flux of mechanical energy in incompressible boundary layer type flow is formed by taking the product of the streamwise velocity and the streamwise momentum equation. In differential form, the momentum and mechanical energy equations are redundant, since one is just a scalar multiple of the other. If the flux of momentum is in balance at each point, then the flux of mechanical energy is also in balance at each point. In the momentum integral formulation, redundancy between the momentum and energy equations does not exist, since the momentum flux is not required to balance at each point. The momentum flux is of course required to balance on the average, but this is not a sufficient condition to insure an average balance of energy flux. In essence, a global conservation of momentum does not imply a global conservation of mechanical energy. An independent equation must be used to enforce an overall balance of mechanical energy.

The equation needed to insure a global balance of mechanical energy is the energy integral equation. In analogy to the momentum integral equation, it is created by integrating the differential form of the energy equation. It has the form

$$\int_0^H \hat{u}\Gamma(\hat{u},\hat{p}) = 0 \tag{5.13}$$

If the velocity and pressure profiles can be specified by two unknown scale functions, then the momentum integral and energy integral equations are sufficient to solve the problem. It should be remarked that other possibilities exist for closing a two-equation system. The momentum integral equation along with a "moment of momentum" equation have been used by previous investigators[12,61]. While this alternate formulation leads to a solution, it should be criticized in that no attempt is made to conserve energy. When a choice exists, the energy integral equation should be preferred over other possible equations that lack physical meaning.

In situations where more than two scale functions must be determined, the momentum integral and energy integral do not provide enough information to close the system. This presents an apparent dilemma, since all three invariants of the flow (mass, momentum, and energy) have already been specified. No further equations which impose physical constraints on the system may be formulated. There is danger in imposing some non-physical condition, since this may overdetermine the system. A way out of this difficulty is to restate the energy equation in an approximate form. This operation then leads to additional conditions that require the error made in the approximation be minimized.

The approximate energy integral equation is derived as follows. Suppose that the function $\hat{u}$ can be decomposed in terms of a suitable set of basis functions. Then it is permissible to write

$$\hat{u} = \sum_{i=1}^{\infty} a_i(x) w_i(y) \tag{5.14}$$

As an approximation, assume that the $\hat{u}$ which multiplies Γ in the energy integral equation can be represented by a finite number of terms in this series. The $\hat{u}$ that appears in the operator Γ itself is not expanded, but is left intact. In this case the energy integral equation (Eq. (5.13)) is approximated by

$$\sum_{i=1}^{N} a_i(x) \int_0^H w_i(y)\Gamma(\hat{u},\hat{p})dy = 0 \tag{5.15}$$

The sum is made to vanish by imposing the strong condition that each of its components vanish independently. This requirement yields the following sequence of equations

$$\int_0^H w_i(y)\Gamma(\hat{u},\hat{p})dy = 0 \qquad i = 1, 2, ..., N \tag{5.16}$$

Note the similarity between this equation and the momentum integral equation (Eq. (5.12)). The two are not independent, since the function 1 that weights Γ in the momentum integral equation is either contained directly in the basis w_i, or can be generated as a linear combination of these. If the basis functions are chosen so that $w_1 = 1$, then the momentum integral equation is actually the first term approximation to the energy integral equation[1]. In this case Eq. (5.16) alone is sufficient to insure a global conservation of momentum and an approximate global conservation of mechanical energy. In this work the basis functions are always chosen so that this condition is satisfied.

At this point it is worthwhile to reinterpret Eq. (5.16) as a statement of the weighted residual method[63]. The term $\Gamma(\hat{u},\hat{p})$ represents the residual error left when the approximate velocity and pressure profiles are substituted into the momentum equation. The basis w_i can be thought of as a set of weighting functions. With these interpretations, Eq. (5.16) states that each projection of the error on the finite space spanned by the weighting functions, w_i vanishes. The fact that the error is orthogonal to all the members of w_i implies that it is *minimized* with respect to these functions. In the limit as infinitely many projections are taken, the error will be driven to zero everywhere. This follows from the fact that the only function that is orthogonal to all members of a complete set is the function zero itself.

The weighting functions are yet unspecified. The only restrictions imposed on these are that they form a complete set and that $w_1 = 1$. In most cases the weighting functions are chosen to make the integrations as easy as possible. In some cases

[1]This fact may explain why previous investigators[12,61] obtained reasonable results without explicitly enforcing the energy integral equation

it is possible to choose the weighting functions such that the approximate solution converges to the exact one in an optimal way. Weighting functions from both of these categories are used in this work. More discussion concerning the individual sets of weighting functions will be discussed in a later section.

Let us now return to Eq. (5.16) and see how it provides a set of equations for the scale functions. First consider the residual. If the approximate solutions $\hat{u}$ and $\hat{p}$ are substituted into Eq. (5.9), the right hand side no longer vanishes, but rather will equal some residual error, ϵ

$$\begin{aligned} \epsilon &= \Gamma\{\hat{u},\hat{p}\} \\ &= \Lambda\{\hat{u},\hat{p}\} - \frac{1}{\rho}\frac{\partial \tau}{\partial y} \\ &= \left[\hat{u}\frac{\partial \hat{u}}{\partial c_j} - \frac{\partial \hat{u}}{\partial y}\int_{y_1}^{y}\frac{\partial \hat{u}}{\partial c_j}dy + \frac{1}{\rho}\frac{\hat{p}}{\partial c_j}\right]\dot{c}_j - \frac{1}{\rho}\frac{\partial \tau}{\partial y} \end{aligned} \tag{5.17}$$

Note that the residual is linear in the first derivatives of the scale functions. For convenience the residual may be written more compactly as

$$\epsilon(x,y) = q_j\dot{c}_j(x) - \frac{1}{\rho}\frac{\partial \tau}{\partial y} \tag{5.18}$$

Now if the above form for the residual is substituted into Eq. (5.16), the following system of equations for the scale functions arises

$$\dot{c}_j\int_0^H w_i q_j dy = \frac{1}{\rho}\int_0^H w_i\frac{\partial \tau}{\partial y}dy \qquad i = 1,2,...,N \tag{5.19}$$

The right hand side is simplified through integration by parts. If the shear due to the boundary layer at the wall is neglected, the above system of equations may be written as

$$\dot{c}_j\int_0^H w_i q_j dy = -\int_0^H \frac{\partial w_i}{\partial y}\frac{\tau}{\rho}dy \tag{5.20}$$

This system of equations for the scale functions may be written more compactly in matrix form as

$$A_{ij}\dot{c}_j = b_i \tag{5.21}$$

where

$$A_{ij} = \int_0^H w_i q_j dy \tag{5.22}$$

and

$$b_i = -\int_0^H \frac{\partial w_i}{\partial y} \frac{\tau}{\rho} dy \tag{5.23}$$

Since both the shape functions, $q_j(c_k(x), y)$, and the weighting functions $w_i(y)$ are universal, the integrations can be done once and for all. The resulting matrix and right hand side depend only on the values of the scale functions. The solution is obtained by marching the above system of equations downstream, computing and inverting the matrix at each step.

5.4 Quasi Self-Preserving Velocity Profiles

The approximate velocity profiles used here are formed by making minor modifications to the self-similar profiles observed for free jets. These modifications involve a generalization of the evolution of scaling parameters such as the centerline velocity and the jet half-width. In the self-similar solution, the scaling parameters are rigidly defined functions of the streamwise distance, while in the approximate profiles, these quantities are taken to be general functions of x. As an example, experiments for a two-dimensional free jet[64, page 21] give the following self-similar velocity profile

$$u(x, y) = 3.5 \frac{u_{ex}}{\sqrt{x/t}} \exp\left[-0.693\left(\frac{y}{x/10}\right)^2\right] \tag{5.24}$$

where u_{ex} is the jet exit velocity, and t is the jet exit width. In this expression the centerline velocity decays like the inverse square root of x, while the characteristic width of the jet grows linearly with x. The approximate velocity profiles are made more flexible than this by allowing the scaling parameters to vary with x in a general sense. The above velocity profile is modified accordingly as follows

$$\hat{u}(x, y) = u_0(x) + u_1(x) \exp\left[-\left(\frac{\alpha y}{b(x)}\right)^2\right] \tag{5.25}$$

where the scale factors u_0, u_1, and b are functions of x to be determined in the solution process. The shape of the approximate velocity profile is the same as a self

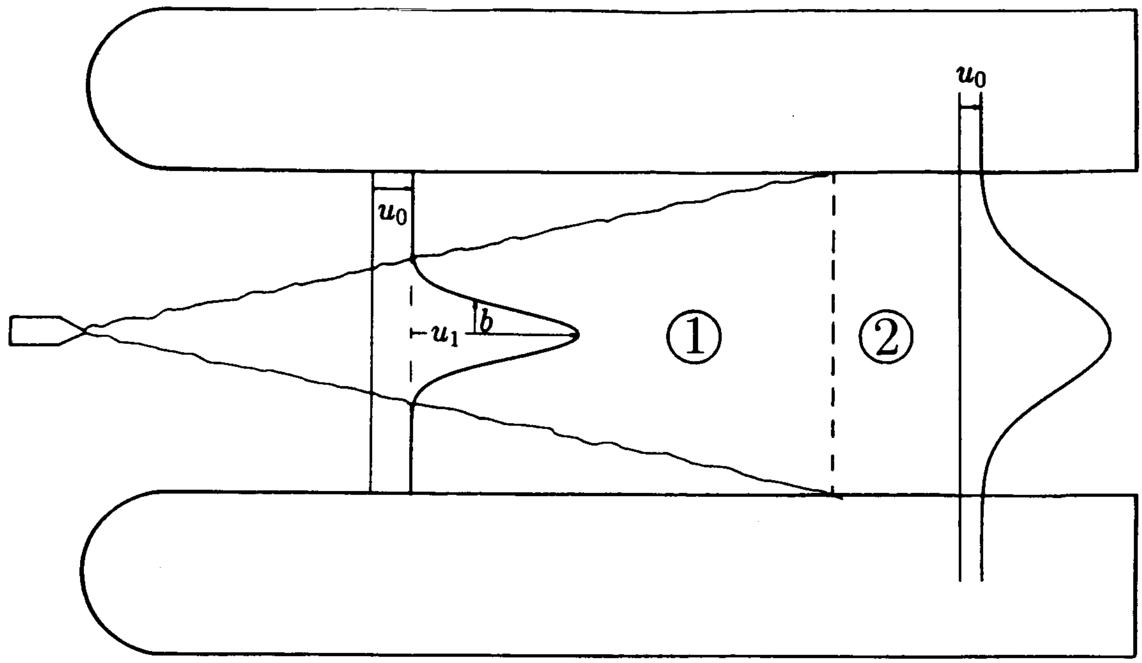


Figure 5.1: Universal nature of the velocity profile. The pressure is related to u_0 through Bernoulli's equation only in region 1. In region 2 u_0 is a fictitious quantity, and the pressure must be computed directly from the momentum equation.

similar profile, but the evolution of its scale is not restricted to obey the rules for mathematical similarity. For this reason the profiles are called quasi self-preserving.

An important feature of the approximate velocity profile is that it is valid from the jet nozzle all the way to the shroud exit. The velocity profile at each cross-section within the ejector walls is assumed to be the central portion of the velocity profile of an effective jet which develops in an unbounded space. The effective jet is special in the sense that it only satisfies the conservation laws in the region bounded by the channel walls. This idea was suggested by Abramovich[48, page 634], who noticed that experimental data could be rationalized in this way. Figure 5.1 shows the basic idea. The earliest attempts at using the integral method to solve confined jets did not make use of this type of formulation. Consequently, the solutions obtained were either restricted to the inlet region of the duct[61], or unnecessarily complicated by the inclusion of two separate expressions for the velocity profile[62].

A certain amount of confusion is evident in the literature on how to properly account for the pressure when using the unified velocity profile formulation. To understand the source of difficulty, consider the two regions shown in the sketch of Figure 5.1. In region 1 the real jet and the effective jet are actually the same. Fluid with velocity u_0, yet untouched by the primary jet, can be related to the pressure through Bernoulli's equation. That is $p = p_{atm} - 1/2\rho u_0^2$. This relation is used in region 1 to eliminate the pressure in terms of the external velocity. In region 2 the viscous flow extends all the way across the channel. The velocity profile within this region has the shape of the middle portion of the effective jet which does not acknowledge the presence of the walls. The quantity u_0 no longer has a physical meaning, but rather is a fictitious quantity that represents the external velocity in the effective profile. The evolution of the velocity profile in region 2 is determined by applying the conservation laws to only that portion of the flow contained within the channel walls. Since it is only the region within the ejector that is required to satisfy the conservation laws, there is not a direct connection between the pressure within the ejector and the fictitious inviscid velocity outside. Application of Bernoulli's equation to relate the pressure to u_0 in region 2 does not make sense, since it would imply that the pressure within the ejector is governed by the fictitious jet profile and not the properties of the flow within the ejector. In spite of this, there are instances in the literature where Bernoulli's equation is used in this region[54]. The correct way to handle the pressure in region 2 is to include it as an independent unknown quantity in the solution of the momentum equation.

5.5 Eddy Viscosity Hypothesis

The Boussinesq approximation for the turbulent stresses was introduced in Section 5.1. The eddy viscosity coefficient contained in this relation is determined from a suitable Reynolds stress model. In this case a simple algebraic model is used. From dimensional considerations it is evident that the eddy viscosity coefficient is composed of the product of a length and a velocity, that is

$$\nu_t \sim u_t l_t \tag{5.26}$$

where u_t and l_t are the characteristic eddy velocity and eddy size respectively. These quantities are not known, but can be estimated from the properties of the mean flow. In this work, the following scaling hypothesis is used

$$\nu_t = k u_1 b \tag{5.27}$$

where u_1 is the jet excess velocity and b is the jet excess velocity half-width. Experimental measurements of the Reynolds stresses[47] support this scaling. Tavella[12], who used this same scaling, obtained close agreement with experiments. Tavella determined the constant k by assuming that the spreading rate of the confined jet should reduce to that of a free jet in the close neighborhood of the jet origin. This analysis results in a value of $k = 0.0283$. This value is adopted in the present work as well.

5.6 Viscous Solution for the Single-Jet Ejector

For the one jet case, the velocity profile used by Tavella and Roberts [12] is adopted

$$\hat{u}(x,y) = u_0(x) + u_1(x)\exp(\eta) \tag{5.28}$$

where

$$\eta = \frac{\alpha y}{b(x)} \tag{5.29}$$

The constant α is chosen to be $\sqrt{\ln(2)}$, so that b has the interpretation of the excess velocity $(u - u_0)$ half-width.

In order to justify the use of this profile, Tavella[12] performed a statistical analysis in which the assumed profile shape was compared against experimental data for a confined turbulent jet. He found that the profile fit the data exceptionally well near the nozzle, but degraded slightly toward the end of the channel. In the worst case, however, the fit was still within the scatter of the data. Tavella also tried a more flexible profile in which the exponent was developed in powers of η. This representation did not produce any significant improvement in accuracy, and thus was abandoned in favor of the simpler expression.

The flow variables are symmetric with respect to the jet centerline in the single-jet case, since the jet is issued along the channel symmetry plane. As a result, the jet centerline is confined to remain on the plane of symmetry, and thereby follows a straight trajectory. The radius of curvature of the jet centerline is infinite in this instance, and the transverse momentum equation (Eq. (5.3)) reduces to

$$\frac{\partial p}{\partial y} = 0 \tag{5.30}$$

which implies that the pressure is a function of x alone. Thus

$$\hat{p}(x,y) = \bar{p}(x) \tag{5.31}$$

5.6.1 Matching Region

Within the viscous-inviscid matching region, the external velocity and the pressure are known from the inviscid solution. With $\mathring{u}_0$ and $\mathring{p}$ known, the viscous problem reduces to finding solutions for u_1 and b. The momentum integral and energy integral equations are use to solve for these two unknowns.

The derivation is simpler if the momentum and energy integral equations are manipulated slightly before substitution of the approximate velocity and pressure profiles. The momentum and energy integral equations may both be written in the following general form

$$\int_0^H \left\{ u^{n+1}\frac{\partial u}{\partial x} + u^n v\frac{\partial u}{\partial y} + u^n\frac{1}{\rho}\frac{\partial p}{\partial x} - u^n\frac{1}{\rho}\frac{\partial \tau}{\partial y} \right\} dy = 0 \tag{5.32}$$

where $n = 0$ for the momentum integral equation and $n = 1$ for the energy integral equation. After algebraic manipulation and use of the continuity equation, the above relation may be written equivalently as

$$\int_0^H \left\{ \frac{\partial}{\partial x}\left[u\left(\frac{1}{n+1}u^{n+1} + u^{n-1}\frac{1}{\rho}p\right)\right] + \frac{\partial}{\partial y}\left[v\left(\frac{1}{n+1}u^{n+1} + nu^{n-1}\frac{1}{\rho}p\right)\right] - \right.$$
$$\left. v\left[nu^{n-1}\frac{\partial p}{\partial y} + n(n-1)u^{n-2}p\frac{\partial u}{\partial y}\right] - \frac{1}{\rho}u^n\frac{\partial \tau}{\partial y} \right\} dy = 0 \tag{5.33}$$

If the upper limit of integration is held fixed, the differentiation with respect to x may be brought outside the integral. The integrals of the derivatives with respect to y are evaluated assuming that $v(0) = 0$, $\tau(0) = 0$, and $\tau(H) = 0$. Finally $\frac{\partial p}{\partial y}$ is assumed to be zero and Eq. (5.5) is used to rewrite the turbulent stress in terms of the mean flow quantities. The momentum and energy integral equations then become

$$\frac{\partial}{\partial x}\int_0^H \left[u^2 + \frac{1}{\rho}p\right] dy + u(H)v(H) = 0 \tag{5.34}$$

$$\frac{\partial}{\partial x}\int_0^H \left[u\left(\frac{1}{2}u^2 + \frac{1}{\rho}p\right)\right] dy + \left(\frac{1}{2}u^2(H) + \frac{1}{\rho}p\right) v(H) = -\nu_t \int_0^H \left(\frac{\partial u}{\partial y}\right)^2 dy \tag{5.35}$$

Bernoulli's equation is valid for the inviscid portion of the inlet flow. Assuming the vertical component of velocity to be small when compared with the horizontal component, the pressure may be related to the external velocity as follows

$$\frac{1}{\rho}\bar{p} = \frac{1}{\rho}p_{atm} - \frac{1}{2}u_0^2 \tag{5.36}$$

Now Eq. (5.28) for the velocity profile, Eq. (5.36) for the pressure profile, and Eq. (5.27) for the eddy viscosity coefficient are substituted into the momentum and energy integral equations (Eqs. (5.34) and (5.35)). The integrations and differentiations are carried out while assuming that $b \ll H$. After some simplification, the following system of equations results

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{b} \end{Bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{Bmatrix} 1 \\ \dot{u}_0 \end{Bmatrix} \tag{5.37}$$

where

$$\begin{aligned} a_{11} &= u_0 + \sqrt{2}u_1 \\ a_{12} &= \tfrac{u_1}{b}\left(u_0 + \tfrac{\sqrt{2}}{2}u_1\right) \\ a_{21} &= \sqrt{2}u_0^2 + 3u_0u_1 + \sqrt{\tfrac{3}{2}}u_1^2 \\ a_{22} &= \tfrac{u_1}{b}\left(\sqrt{2}u_0^2 + \tfrac{3}{2}u_0u_1 + \tfrac{1}{2}\sqrt{\tfrac{2}{3}}u_1^2\right) \\ b_{11} &= 0 \\ b_{12} &= -2u_1 \\ b_{21} &= -\tfrac{k\alpha^2u_1^3}{b} \\ b_{22} &= -u_1\left(2\sqrt{2}u_0 + \tfrac{3}{2}u_1\right) \end{aligned} \tag{5.38}$$

5.6.2 Fully Viscous Region

Within the fully viscous region of the ejector, the external velocity and the pressure are no longer able to be computed from the inviscid solution. The system of viscous equations must be enlarged so that u_0 and $\bar{p}$ may also be obtained. With the addition of two more unknowns, the momentum integral and energy integral equations alone do not provide enough information to close the system.

One additional equation is derived from the condition that no flow pass through the ejector wall. This condition is stated as

$$v(x, y = H) = u(x, y = H)\frac{dH}{dx} \tag{5.39}$$

When the above boundary condition is enforced, the system is still one equation short of closure. Closure is obtained through use of the weighted residual method.

The first step in implementing the weighted residual method is to derive the individual terms in the momentum equation from the approximate velocity and pressure profiles. Equations. (5.8), (5.28), (5.31), (5.5), and (5.27) are used to give

$$\frac{\partial \hat{u}}{\partial x} = \dot{u}_0 + \exp(\eta)\dot{u}_1 + 2\frac{u_1}{b}\eta^2\exp(\eta)\dot{b} \tag{5.40}$$

$$\hat{v} = -\frac{b}{\alpha}\left[\eta\dot{u}_0 + \frac{\sqrt{\pi}}{2}\mathrm{erf}(\eta)\dot{u}_1 + \frac{u_1}{b}\left(\frac{\sqrt{\pi}}{2}\mathrm{erf}(\eta) - \eta\exp(\eta)\right)\dot{b}\right] \tag{5.41}$$

$$\frac{\partial \hat{u}}{\partial y} = -2\frac{\alpha}{b}u_1\eta\exp(\eta) \tag{5.42}$$

$$\frac{\partial \hat{p}}{\partial x} = \dot{\bar{p}} \tag{5.43}$$

The residual is now constructed according to Eq. (5.17)

$$\begin{aligned}\Lambda\{\hat{u}, \hat{p}\} &= q_j\dot{c}_j \\ &= \left[u_0 + u_1(1 + 2\eta^2)\exp(\eta)\right]\dot{u}_0 + \\ &\quad \left[u_0\exp(\eta) + 2u_1\left(\frac{\sqrt{\pi}}{2}\eta\exp(\eta)\mathrm{erf}(\eta) + \frac{1}{2}\exp(-2\eta^2)\right)\right]\dot{u}_1 + \\ &\quad \dot{\bar{p}} +\end{aligned} \tag{5.44}$$

$$2\frac{u_1}{b}\left[u_0\eta^2\exp(\eta) + \frac{\sqrt{\pi}}{2}u_1\eta\exp(\eta)\text{erf}(\eta)\right]\dot{b}$$

$$\frac{\tau}{\rho} = -2k\alpha u_1^2\eta\exp(\eta) \tag{5.45}$$

The weighting functions are chosen primarily for algebraic convenience. They are simply the power sequence

$$w_j = y^{j-1} \qquad j = 1, 2, 3 \tag{5.46}$$

The choice of weighting functions allows the integrations indicated by Eqs. (5.22) and (5.23) to be performed analytically. When the integrals are evaluated, the system of equations may be written as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{Bmatrix} \dot{u}_0 \\ \dot{u}_1 \\ \dot{p} \\ \dot{b} \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{Bmatrix} \tag{5.47}$$

The first three equations above are formed by weighting the residual with y^0, y^1, and y^2 respectively. The fourth equation enforces the flow tangency boundary condition at the wall. With the definitions

$$\begin{aligned} \eta_H &= \frac{\alpha H}{b} \\ E_1 &= \exp(-\eta_H^2) \\ E_2 &= \exp(-2\eta_H^2) \\ F_1 &= \frac{\sqrt{\pi}}{2}\text{erf}(\eta_H) \\ F_2 &= \frac{1}{2}\sqrt{\frac{\pi}{2}}\text{erf}(\sqrt{2}\eta_H) \end{aligned}$$

the individual terms in Eq. (5.47) are

$$a_{11} = \frac{b}{\alpha}\left[\left(u_0 - u_1E_1\right)\eta_H + 2u_1F_1\right]$$

$$
\begin{aligned}
a_{12} &= \frac{b}{\alpha}\left[\left(u_0 - u_1 E_1\right) F_1 + 2u_1 F_2\right] \\
a_{13} &= \frac{b}{\alpha}\left[\eta_H\right] \\
a_{14} &= \frac{b}{\alpha}\frac{u_1}{b}\left[\left(u_0 - u_1 E_1\right)\left(F_1 - \eta_H E_1\right) + u_1\left(F_2 - \eta_H E_2\right)\right] \\
a_{21} &= \frac{1}{2}\left(\frac{b}{\alpha}\right)^2\left[u_0\eta_H^2 + u_1\left(3(1-E_1) - 2\eta_H^2 E_1\right)\right] \\
a_{22} &= \frac{1}{2}\left(\frac{b}{\alpha}\right)^2\left[u_0(1-E_1) + u_1\left((1-E_2) + F_1(F_1 - 2\eta_H E_1)\right)\right] \\
a_{23} &= \frac{1}{2}\left(\frac{b}{\alpha}\right)^2\left[\eta_H^2\right] \\
a_{24} &= \frac{1}{2}\left(\frac{b}{\alpha}\right)^2\frac{u_1}{b}\left[2u_0\left(1 - (1+\eta_H^2)E_1\right) + u_1\left(F_1(F_1 - 2\eta_H E_1) + \frac{1}{2}(1-E_2)\right)\right] \\
a_{31} &= \left(\frac{b}{\alpha}\right)^3\left[\frac{1}{3}u_0\eta_H^3 + u_1\left(2(F_1 - \eta_H E_1) - \eta_H^3 E_1\right)\right] \\
a_{32} &= \left(\frac{b}{\alpha}\right)^3\left[\frac{1}{2}u_0\left(F_1 - \eta_H E_1\right) + u_1\left(\frac{1}{2}(F_2 - \eta_H E_2) + F_2 - (1+\eta_H^2)E_1 F_1\right)\right] \\
a_{33} &= \left(\frac{b}{\alpha}\right)^3\left[\frac{1}{3}\eta_H^3\right] \\
a_{34} &= \left(\frac{b}{\alpha}\right)^3\frac{u_1}{b}\left[u_0\left(\frac{3}{2}(F_1 - \eta_H E_1) - \eta_H^3 E_1\right) + u_1\left(-(1+\eta_H^2)E_1 F_1 + F_2 + \frac{1}{4}(F_2 - \eta_H E_2)\right)\right] \\
a_{41} &= \frac{b}{\alpha}\left[\eta_H\right] \\
a_{42} &= \frac{b}{\alpha}\left[F_1\right] \\
a_{43} &= 0 \\
a_{44} &= \frac{b}{\alpha}\frac{u_1}{b}\left[F_1 - \eta_H E_1\right]
\end{aligned}
$$

$$
\begin{aligned}
b_1 &= 0 \\
b_2 &= \frac{1}{2}\left(\frac{b}{\alpha}\right)^2\left[2\frac{k\alpha^2u_1^2}{b}\left(1-E_1\right)\right] \\
b_3 &= \left(\frac{b}{\alpha}\right)^3\left[2\frac{k\alpha^2u_1^2}{b}\left(F_1-\eta_H E_1\right)\right] \\
b_4 &= -\left(\frac{b}{\alpha}\right)^3\left[\frac{\alpha}{b}\frac{dH}{dx}(u_0+u_1E_1)\right]
\end{aligned}
$$

5.7 Viscous Solution for the Dual-Jet Ejector

The velocity profile for the two jet case is constructed from interfering hyperbolic functions

$$
\begin{aligned}
\hat{u}(x,y) &= u_0(x)+\frac{1}{2}a(x)\left[\tanh(\eta+\eta_1)-\tanh(\eta-\eta_1)\right]+ \\
& \quad u_1(x)\left[\operatorname{sech}^2(\eta+\eta_1)+\operatorname{sech}^2(\eta-\eta_1)\right] \qquad (5.48)
\end{aligned}
$$

where

$$\eta = \frac{\alpha y}{b(x)} \qquad (5.49)$$

and

$$\eta_1 = \frac{\alpha y_1(x)}{b(x)} \qquad (5.50)$$

The shape of the profile is shown in Figure 5.2. The parameter $y_1(x)$ represents the location of the jet centerline, $u_0(x)$ the external velocity at the ejector wall, $u_0(x)+a(x)$ the external velocity at the channel centerline, $u_1(x)$ the jet excess velocity, and $b(x)$ the jet excess velocity half-width. The hyperbolic secant functions are patterned after the self-similar solution to a free jet [64, page 19]. The hyperbolic tangent functions are used to allow for unequal secondary flow velocity on either side of the jets. In this case the constant $\alpha = \cosh^{-1}(\sqrt{2})$.

In the two jet case, the lack of symmetry in the secondary flow with respect to the jet centerline allows a pressure difference to develop across the jets. The pressure is therefore not constant within the layer, but rather develops some profile

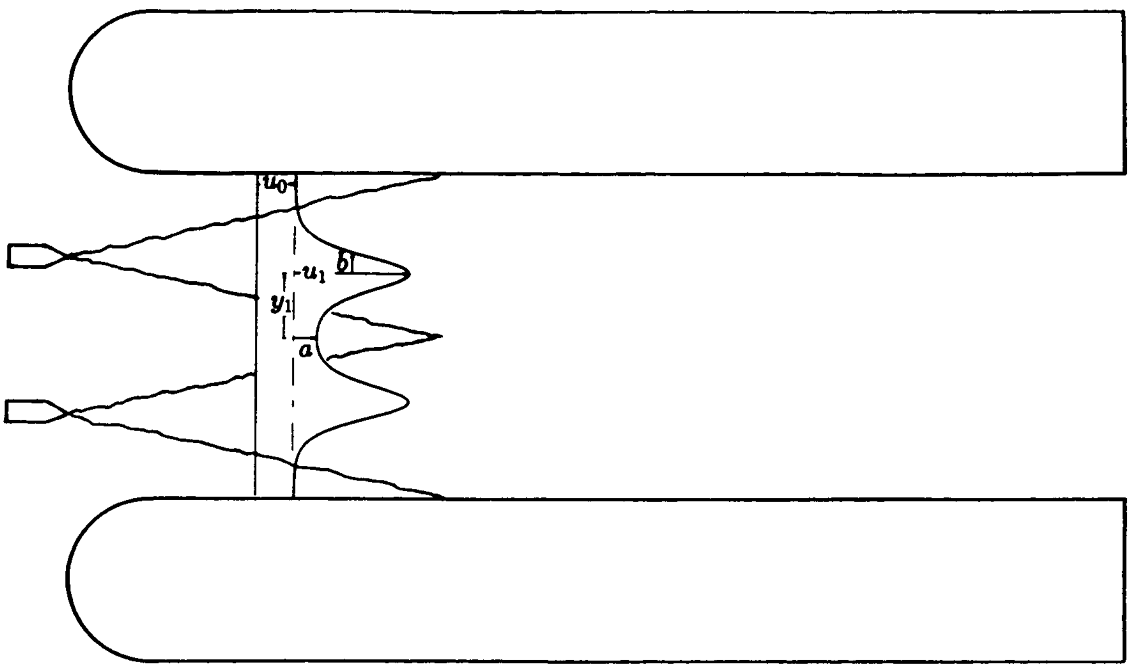


Figure 5.2: Velocity profile for the dual-jet ejector

in making the transition from the external pressure on either side. An approximate expression for this profile that satisfies Bernoulli's equation on either side of the jet is

$$\hat{p}(x,y) = \bar{p} - \frac{1}{2}\rho a(u_0 + 1/2a)\left[\tanh(\eta + \eta_1) - \tanh(\eta - \eta_1) - 1\right] \tag{5.51}$$

where the average pressure $\bar{p}$ is defined as

$$\bar{p} = p_{atm} - \frac{1}{2}\rho\left(u_0^2 + u_0 a + 1/2a^2\right) \tag{5.52}$$

5.7.1 Matching Region

Within the viscous-inviscid matching region, the jet centerlines follow curved trajectories as a result of the non-uniform pressure field developed by the secondary flow. The boundary layer equations still apply in the case of a moderately curved layer provided they are written in a curvilinear coordinate system. If s and n are the directions of a curvilinear coordinate system that is locally tangent to the jet centerline, the boundary layer equations may be recast as follows

$$\frac{\partial u}{\partial s} + \frac{\partial v}{\partial n} = 0 \tag{5.53}$$

$$u\frac{\partial u}{\partial s} + v\frac{\partial u}{\partial n} + \frac{1}{\rho}\frac{\partial p}{\partial s} = \frac{1}{\rho}\frac{\partial \tau}{\partial n} \tag{5.54}$$

$$\frac{\partial p}{\partial n} = \frac{\rho u^2}{R} \tag{5.55}$$

The velocity and pressure profiles must also be recast in the curvilinear coordinate system. In considering the jet that lies in the upper half-plane, the following transformation is used to rewrite the expressions for the velocity and pressure profiles in terms of a coordinate system that is everywhere tangent to the jet centerline

$$x \rightarrow s \tag{5.56}$$

$$y - y_1 \rightarrow n \tag{5.57}$$

With the assumption that $y + y_1 \gg b$, the velocity and pressure profiles become

$$\hat{u}(s,n) = u_0(s) + \frac{1}{2}a(s)\Big[1 - \tanh(\zeta)\Big] + u_1(s)\mathrm{sech}^2(\zeta) \tag{5.58}$$

$$\hat{p} = \bar{p}(s) + \frac{1}{2}\rho a(s)\Big(u_0(s) + 1/2a(s)\Big)\tanh(\zeta) \tag{5.59}$$

where

$$\zeta = \frac{\alpha n}{b} \tag{5.60}$$

Within the viscous-inviscid matching region u_0, a, and $\bar{p}$ can be deduced from the inviscid solution. The viscous problem therefore reduces to finding solutions for u_1, b, and y_1. As in the single-jet case, the momentum and energy integral equations are used to provide equations for u_1 and b. An equation for y_1 is derived from the normal momentum equation.

Due to lack of symmetry with respect to the jet centerline in the dual-jet case, the integrals in the momentum and energy integral equations must extend on both sides of the jet centerline. Analogous to Eqs. (5.34) and (5.35), the momentum and energy integral equations for the dual-jet ejector written in the curvilinear coordinate system are

$$\frac{\partial}{\partial s}\int_{-H/2}^{H/2}\left[u^2+\frac{1}{\rho}p\right]dn+u(H/2)v(H/2)-u(-H/2)v(-H/2)=0 \tag{5.61}$$

$$\begin{aligned}&\frac{\partial}{\partial s}\int_{-H/2}^{H/2}\left[u\left(\frac{1}{2}u^2+\frac{1}{\rho}p\right)\right]dn-\frac{1}{\rho}\int_{-H/2}^{H/2}v\frac{\partial p}{\partial n}dn+\\&v(H/2)\left(\frac{1}{2}u^2(H/2)+\frac{1}{\rho}p(H/2)\right)-v(-H/2)\left(\frac{1}{2}u^2(-H/2)+\frac{1}{\rho}p(-H/2)\right)\\&=-\nu_t\int_{-H/2}^{H}/2\left(\frac{\partial u}{\partial n}\right)^2 dn\end{aligned} \tag{5.62}$$

Calculation of the Jet Trajectory

As stated in Eq. (5.3) the pressure difference acting across the jets results in a curvature of their centerlines. Since the pressure difference across the jet is known from the inviscid solution, Eq. (5.55) may be integrated across the jet to yield an expression for the curvature of the jet centerline

$$\kappa \equiv \frac{1}{R}=\frac{\Delta p}{J} \tag{5.63}$$

where

$$\Delta p=\frac{1}{2}\rho a(2u_0+a) \tag{5.64}$$

$$J=\rho\int_{-2.5b}^{2.5b}u^2 dn \tag{5.65}$$

The curvature of the jet centerline is also related to the derivatives of $y_1(x)$

$$\kappa=\frac{\ddot{y}_1}{(1+\dot{y}_1^2)^{3/2}} \tag{5.66}$$

With the definition $q=\dot{y}_1$, the above equation may be written as the following two first order differential equations that govern the jet trajectory

$$\begin{aligned}\dot{y}_1 &= q\\ \dot{q} &= \kappa(1+q^2)^{3/2}\end{aligned} \tag{5.67}$$

These two equations together with Eq. (5.63) are integrated along with the rest of the equations for the viscous solution.

System of Equations

Equations. (5.58) and (5.59) for the velocity and pressure profiles respectively are substituted into the momentum and energy integral equations (Eqs. (5.61) and (5.62)) and the integrations and differentiations carried out assuming $H/2 \gg b$. The results of these operations are combined with Eq. (5.67) to give

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{b} \\ \dot{y}_1 \\ \dot{q} \end{Bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & 0 & 0 \\ b_{41} & 0 & 0 \end{bmatrix} \begin{Bmatrix} 1 \\ \dot{u}_0 \\ \dot{a} \end{Bmatrix} \tag{5.68}$$

where

$$a_{11} = u_0 + \frac{4}{3}u_1 + \frac{1}{2}a$$

$$a_{12} = \frac{1}{b}\left[u_1\left(u_0 + \frac{2}{3}u_1 + \frac{1}{2}a\right) - \frac{1}{4}\left(1 - \frac{\ln 2}{2}\right)a^2\right]$$

$$a_{21} = 2u_0\left(u_0 + 2u_1 + a\right) + u_1\left(\frac{8}{5}u_1 + 2a\right) + \frac{1}{2}a^2$$

$$a_{22} = \frac{1}{b}\left\{u_1^2\left(2u_0 + \frac{8}{15}u_1 + a\right) + 2u_0u_1\left(u_0 + a\right) + \frac{1}{2}a^2\left[-\left(1 - \frac{\ln 2}{2}\right)\left(u_0 + \frac{1}{2}a\right) + u_1\right]\right\}$$

$$b_{11} = 0$$

$$b_{12} = -2u_1$$

$$b_{13} = -u_1 + \frac{1}{2}\left(1 - \frac{\ln 2}{2}\right)a$$

$$b_{21} = -\frac{k\alpha^2 u_1}{b}\left(\frac{16}{15}u_1^2 + \frac{1}{3}a^2\right)$$

$$b_{22} = -2u_1\left(2u_0 + u_1 + a\right) + \frac{1}{4}a^2$$

$$b_{23} = -u_1\left(u_1 + 2u_0 + a\right) + \left(1 - \frac{\ln 2}{2}\right)a\left(u_0 + \frac{1}{2}a\right) + \frac{1}{8}a^2$$

$$b_{31} = q$$

$$b_{41} = \frac{\frac{1}{2}a(2u_0 + a)}{2\frac{b}{\alpha}\left[\frac{2}{3}u_1^2 + 2u_0u_1 + u_1a - \frac{1}{4}a^2\right]}\left[1+q^2\right]^{3/2}$$

The vertical component of velocity for the matching region is found from Eqs. (5.53) and (5.58)

$$\begin{aligned} v &= -\int_0^n \frac{\partial u}{\partial s}dn \\ &= -\frac{b}{\alpha}\left\{\zeta\dot{u}_0 + \tanh(\zeta)\dot{u}_1 - \frac{1}{2}\left(\ln\cosh(\zeta) - \zeta\right)\dot{a} + \right. && (5.69) \\ & \left.\frac{1}{b}\left[u_1\zeta\tanh^2(\zeta) + \left(u_1 + \frac{1}{2}a\zeta\right)\tanh(\zeta) - \frac{1}{2}a\ln\cosh(\zeta) - u_1\zeta\right]\dot{b}\right\} && (5.70) \end{aligned}$$

5.7.2 Fully Viscous Region

Within the fully viscous region of the ejector, the inviscid solution no longer provides the information to determine u_0, a, and y_1. As in the single-jet case, the method of weighted residuals is used to generate additional equations for these unknowns. Unlike the single-jet case, however, the integrals that arise in the fully viscous region of the dual-jet ejector are quite difficult to evaluate analytically. For this reason, the integrals are evaluated numerically at each streamwise location. When an efficient Simpson's rule algorithm is used to perform the integrations, the time required to compute the fully viscous portion of the flow is still quite small.

The fully viscous region begins far enough inside the ejector to assume that the pressure has become uniform across the channel. The jet trajectories correspondingly are no longer curved, but rather follow straight trajectories. It is therefore appropriate to return to a cartesian coordinate system. The velocity profile is given by Eq. (5.48) and the pressure profile reduces to

$$\hat{p}(x, y) = \bar{p}(x) \qquad (5.71)$$

The weighting functions are chosen in this instance to minimize the integrated square of the error. Using Eq. (5.18), the integrated square of the error may be written as

$$\int_0^H \epsilon^2 dy = \int_0^H \left[(q_j\dot{c}_j)^2 - 2(q_j\dot{c}_j)\frac{1}{\rho}\frac{\partial\tau}{\partial y} + \left(\frac{1}{\rho}\frac{\partial\tau}{\partial y}\right)^2\right] dy \tag{5.72}$$

Now the integrated square of the error is minimized by requiring that it be stationary with respect to the $\dot{c}_j$.

$$\frac{\partial}{\partial\dot{c}_i}\int_0^H \epsilon^2 dy = 2\int_0^H q_i\left[q_j\dot{c}_j - \frac{1}{\rho}\frac{\partial\tau}{\partial y}\right] dy = 0 \tag{5.73}$$

or after integrating the stress term by parts assuming that $\tau(0) = \tau(H) = 0$

$$\dot{c}_j\int_0^H q_i q_j dy = -\int_0^H \frac{\partial q_i}{\partial y}\frac{\tau}{\rho} dy \tag{5.74}$$

The residual is now constructed from Eqs. (5.48) and (5.71). With the definitions

$$\begin{aligned}
A_1 &= \eta + \eta_1\\
A_2 &= \eta - \eta_1\\
T_1 &= \tanh(\eta + \eta_1)\\
T_2 &= \tanh(\eta - \eta_1)\\
S_1 &= \mathrm{sech}^2(\eta + \eta_1)\\
S_2 &= \mathrm{sech}^2(\eta - \eta_1)\\
Q_1 &= \ln\cosh(\eta + \eta_1)\\
Q_2 &= \ln\cosh(\eta - \eta_1)
\end{aligned}$$

the derivatives of the velocity and pressure profile are

$$\begin{aligned}
\frac{\partial\hat{u}}{\partial x} = {} & \dot{u}_0 + \Big(S_1 + S_2\Big)\dot{u}_1 + \frac{1}{2}\Big(T_1 - T_2\Big)\dot{a} + \\
& \frac{1}{b}\left[\frac{1}{2}a\Big(-A_1S_1 + A_2S_2\Big) + 2u_1\Big(A_1T_1S_1 + A_2T_2S_2\Big)\right]\dot{b} + \\
& \frac{\alpha}{b}\left[\frac{1}{2}a\Big(S_1 + S_2\Big) + 2u_1\Big(-T_1S_1 + T_2S_2\Big)\right]\dot{y}_1
\end{aligned}$$

$$
\begin{aligned}
\hat{v} &= -\frac{b}{\alpha}\Big\{\eta\dot{u}_0 + \Big(T_1+T_2\Big)\dot{u}_1 + \frac{1}{2}\Big(Q_1-Q_2\Big)\dot{a} + \\
&\quad \frac{1}{b}\left[\frac{1}{2}a\Big(-A_1T_1+Q_1+A_2T_2-Q_2\Big)+u_1\Big(-A_1S_1+T_1-A_2S_2+T_2\Big)\right]\dot{b} + \\
&\quad \left[\frac{1}{2}a\Big(T_1+T_2\Big)+u_1\Big(S_1-S_2\Big)\right]\dot{y}_1\Big\} \\
\frac{\partial\hat{u}}{\partial y} &= -\frac{\alpha}{b}\left[\frac{1}{2}a\Big(-S_1+S_2\Big)+2u_1\Big(T_1S_1+T_2S_2\Big)\right] \\
\frac{\partial\hat{p}}{\partial x} &= \dot{\bar{p}}
\end{aligned}
$$

Let the elements of q_j be denoted as q_{u_0}, q_{u_1}, q_p, etc., then

$$
\begin{aligned}
q_{u_0} &= \hat{u} - \frac{b}{\alpha}\eta \\
q_{u_1} &= \Big(S_1+S_2\Big)\hat{u} - \frac{b}{\alpha}\Big(T_1+T_2\Big)\frac{\partial\hat{u}}{\partial y} \\
q_p &= 1 \\
q_a &= \frac{1}{2}\Big(T_1-T_2\Big)\hat{u} - \frac{1}{2}\frac{b}{\alpha}\Big(Q_1-Q_2\Big)\frac{\partial\hat{u}}{\partial y} \\
q_b &= \frac{1}{b}\left[\frac{1}{2}a\Big(-A_1S_1+A_2S_2\Big)+2u_1\Big(A_1T_1S_1+A_2T_2S_2\Big)\right]\hat{u} \\
&\quad -\frac{1}{\alpha}\left[\frac{1}{2}a\Big(-A_1T_1+Q_1+A_2T_2-Q_2\Big)+u_1\Big(-A_1S_1+T_1-A_2S_2+T_2\Big)\right]\frac{\partial\hat{u}}{\partial y} \\
q_{y_1} &= \frac{\alpha}{b}\left[\frac{1}{2}a\Big(S_1+S_2\Big)+2u_1\Big(-T_1S_1+T_2S_2\Big)\right]\hat{u} - \\
&\quad \left[\frac{1}{2}a\Big(T_1+T_2\Big)+u_1\Big(S_1-S_2\Big)\right]\frac{\partial\hat{u}}{\partial y}
\end{aligned}
$$

As in the single-jet case, it is necessary to enforce a flow tangency boundary condition at the ejector wall. The condition is

$$
v(x, y=H) = u(x, y=H)\frac{dH}{dx} \tag{5.75}
$$

The need to enforce this boundary condition requires that one of the equations from the weighted residual method be removed from the system. A bit of experimentation has shown that the conditioning of the system is best if the equation formed with q_{u_0} as the weighting function is replaced with the flow tangency boundary condition. When the integrals are evaluated, the system of equations may be written as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} \begin{Bmatrix} \dot{u}_0 \\ \dot{u}_1 \\ \dot{\bar{p}} \\ \dot{a} \\ \dot{b} \\ \dot{y}_1 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{Bmatrix} \tag{5.76}$$

The first five equations are formed by weighting the residual with q_{u_1}, q_p, q_a, q_b, and q_{y_1} respectively, while the last equation enforces the flow tangency boundary condition at the ejector wall.

Chapter 6

Viscous-Inviscid Matching Procedure

6.1 Iteration Scheme

The goal of the viscous-inviscid matching is to obtain viscous and inviscid solutions that are compatible at their common boundary. Compatibility is achieved when the pressure and velocity fields are continuous at the zonal interface. In order to arrive at compatibility, the viscous and inviscid solutions are allowed to influence each other in an iterative process where information is exchanged at the common boundary. This process simulates the physical viscous-inviscid interaction that is taking place within the ejector. The iterative process must be carefully designed such that it allows each flow region to influence the other, and yet remain both stable and computationally efficient even when the interaction is intense.

In the ejector problem there are two areas where a matching must be done. The first is a matching of velocity and pressure fields along the jet boundary, while the second is a matching of the ejector exit pressure to the atmospheric value. Matching of the flow variables along the jet boundary involves finding the correct distribution of jet entrainment. Matching of the exit pressure is achieved when the value of the primary jet momentum flux is consistent with an assumed value of the ejector inlet pressure. These two matching processes are intertwined, since the value of

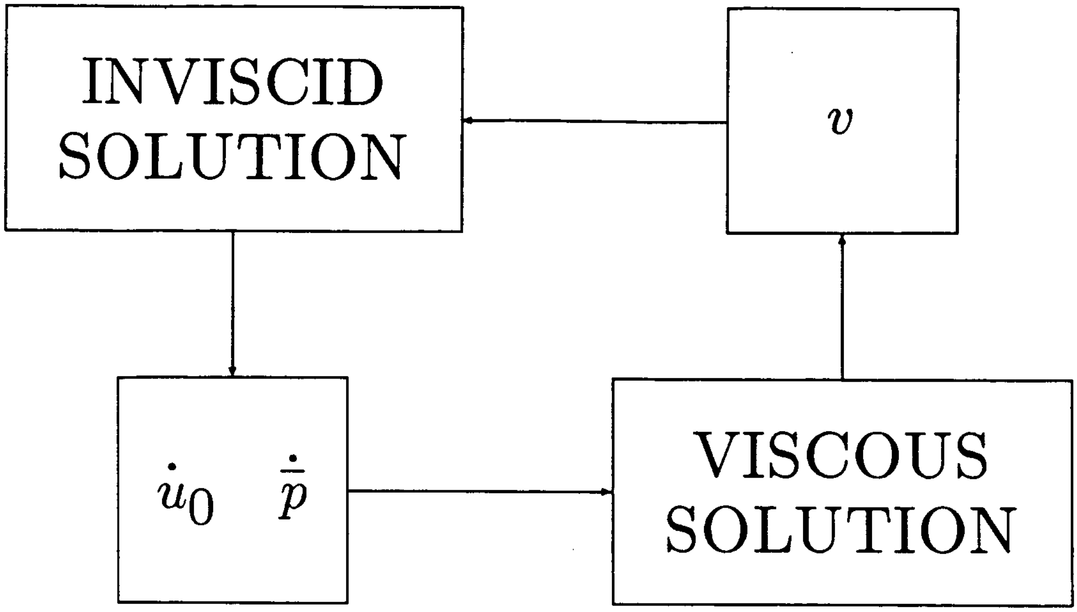


Figure 6.1: Viscous-Inviscid interaction scheme.

the primary jet momentum affects the evolution of the jet, while the interaction between the jet and inviscid flow ultimately affects the ejector exit pressure. The overall iteration scheme is constructed in a nested fashion where an inner loop converges the flow variables at the jet boundary and an outer loop converges the exit pressure.

In the viscous-inviscid loop, the inviscid solution provides both the external velocity, u_0, and the pressure, $\bar{p}$, to be used as boundary conditions in the viscous solution. Once the viscous flow is computed, the jet entrainment velocity is passed to the inviscid region, where it is used to update the suction applied to the panels covering the jet boundary. A new inviscid solution is then calculated, and the cycle repeats. Figure 6.1 illustrates the concept. Convergence is monitored by comparing the distribution of entrainment assumed in the inviscid solution with the actual entrainment computed by the viscous solution.

In the exit pressure matching loop, an initial guess for the primary jet momentum is made. Next the viscous-inviscid matching is performed. The viscous solution is then marched all the way to the ejector exit. The computed exit pressure does not in general agree with the atmospheric value. Accordingly, an adjustment is made to the primary jet momentum flux, and the cycle repeats. Figure 6.2 illustrates the concept. Convergence is achieved in the outer loop when the difference between the exit and atmospheric pressure is negligible.

The matching procedures for both the single and dual jet ejectors are essentially the same. The dual jet case, however, is complicated by the additional interaction which takes place between the two jets. This interaction is manifested both by the effects of jet curvature and by the asymmetric entrainment with respect to the jet centerline. The matching procedures for the one and two jet ejector are described separately below.

6.2 Matching Procedure for the Single-Jet Ejector

6.2.1 Viscous-Inviscid Matching

The ejector flow field is symmetric with respect to the channel centerline. To minimize effort, only the upper half of the of the flow field is solved. The viscous-inviscid matching is therefore contained to the upper half of the boundary between the jet and the inviscid stream. The geometry for the upper half-plane of the inviscid solution was discussed in Section 4.4. The effect of the jet entrainment on the inviscid field is simulated by applying suction to the panels which cover the jet boundary, while the lowered pressure within the inlet is simulated by applying suction to the control station. The matching process determines the distribution of panel suction that makes the viscous and inviscid regions are compatible. The procedure is to iterate between the inviscid horizontal component of velocity and the viscous entrainment. To start the process, an initial guess for the jet entrainment is made, and the panel suction velocities are set accordingly. The inlet suction applied at the control station is a parameter in the exit pressure matching and is assigned a fixed arbitrary value. The inviscid problem is solved, and the velocity components as well as the pressure along the jet boundary are calculated. The quantities $\dot{u}_0$ and

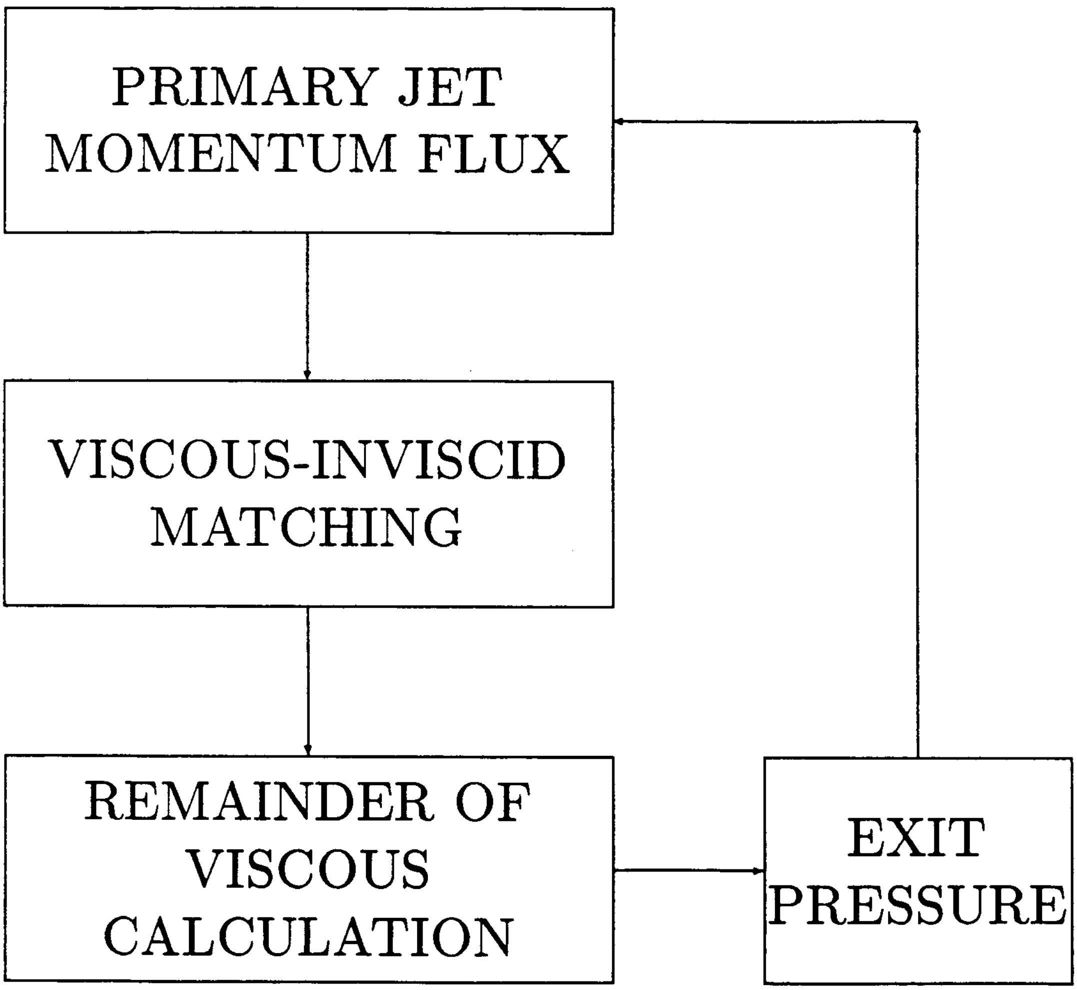


Figure 6.2: Exit pressure matching.

$\dot{\bar{p}}$ are then formed and sent to the viscous region as forcing terms (see Eq. (5.37)). The viscous problem is solved, and the jet entrainment velocity computed via Eq. (5.41). At this point the velocity and pressure fields are compared at the viscous-inviscid interface. The horizontal component of velocity as well as the pressure are already continuous at the interface, since these quantities were extracted from the inviscid solution and transferred directly to the viscous solution through the boundary conditions. The vertical component of velocity, however, will in general not be continuous. Let v_{vis} be the entrainment velocity computed by the viscous solution, v_{inv} be the entrainment velocity computed by the inviscid solution, and V_n the suction velocity applied to the panel where the entrainment is being calculated. Then the following relaxation scheme is used for each panel to produce a correction to its suction velocity

$$V_n^{n+1} = V_n^n + \omega(v_{vis} - v_{inv}) \tag{6.1}$$

Once the correction is made for each of the panels, a new inviscid solution is generated and the whole process repeated.

The parameter ω in Eq. (6.1) is a relaxation factor that is needed to maintain stability. The iteration scheme is only stable if the relaxation factor is allowed to vary with x. The viscous calculation becomes more sensitive to changes in the external inviscid field as the distance from the jet origin is increased. For this reason it is necessary to increase the damping with the streamwise distance. The following linear variation in the relaxation factor is sufficient to control the stability

$$\omega = \left[r - t\left(\frac{x - x_0}{x_{cs} - x_0}\right)\right] \tag{6.2}$$

where

$$r = 1.0, \qquad t = 0.7 \tag{6.3}$$

Here x_0 is the position of the jet nozzle and x_{cs} is the position of the control station, where the viscous-inviscid matching is terminated. While this scheme is under relaxed over most of the jet trajectory, it still converges quite rapidly. Typically four cycles are needed to match the entrainment velocity to within three significant figures. The scheme is also surprisingly robust. No stability problems have been encountered for a wide range of test conditions.

6.2.2 Exit Pressure Matching

Once the viscous-inviscid matching is complete, the remainder of the viscous flow within the channel is computed by marching the system of equations given in Eq. (5.47). The pressure computed at the exit will in general differ from the atmospheric value. An improvement is made by adjusting the primary jet momentum flux.

For a given geometry, the exit pressure depends only on the primary jet momentum flux and the magnitude of the suction applied to the control station. The primary jet momentum flux in turn is specified by an initial velocity, u_{1_0}, while the control station suction is specified by the the velocity u_{cs}. These are the only two relevant velocity scales in the problem. The exit pressure must therefore depend on the ratio u_{1_0}/u_{cs}. Consequently, it is sufficient to vary either one of these quantities while holding the other fixed. It is most convenient to hold u_{cs} fixed and vary just the initial jet velocity u_{1_0}.

A Newton-type iteration is used to converge the exit pressure. First define

$$f^n = (p^n_{exit} - p^n_{atm}) \tag{6.4}$$

$$f'^n = \frac{f^n - f^{n-1}}{u^n_{1_0} - u^{n-1}_{1_0}} \tag{6.5}$$

Then the iteration scheme is

$$u^{n+1}_{1_0} = u^n_{1_0} - \omega f^n \tag{6.6}$$

where

$$\omega = \begin{cases} \omega_0 & n = 1 \\ 1/f'^n & n > 1 \end{cases} \tag{6.7}$$

and where

$$\omega_0 \simeq 0.1 \tag{6.8}$$

Notice that the Newton scheme needs data at two iteration levels. Provision is made for this by incorporating a simple one level scheme for the first step. This iteration scheme converges quite rapidly. Typically four cycles are necessary to converge the exit pressure to the atmospheric value within three significant figures.

The entire iteration process is now complete. A summary of the method is shown schematically in Figure 6.3.

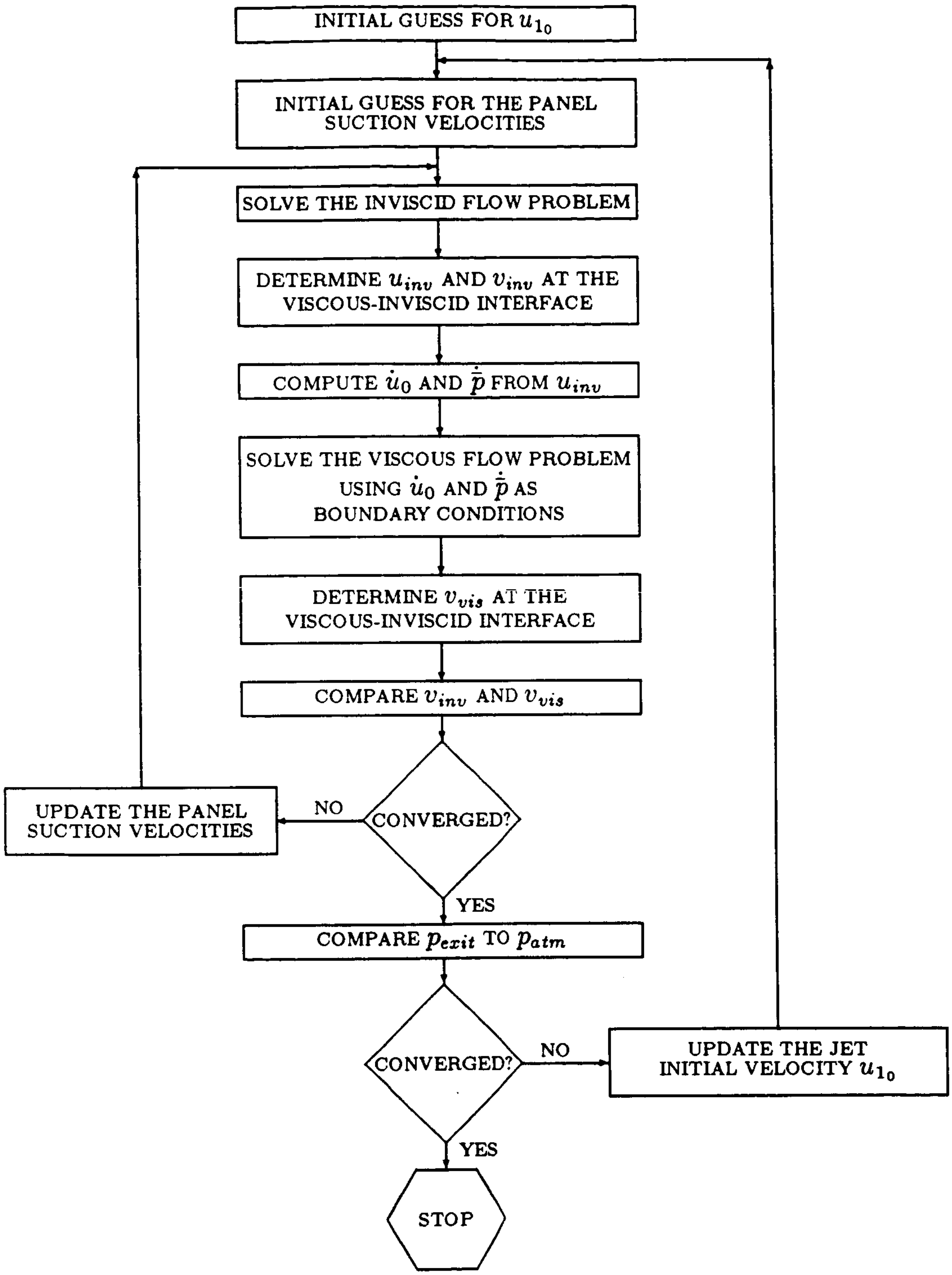


Figure 6.3: Iteration scheme for the single jet ejector. Note how the viscous-inviscid matching loop is nested within the exit pressure matching loop.

6.3 Matching Procedure for the Dual-Jet Ejector

The matching procedure for the dual-jet ejector is conceptually the same as the single-jet case. The procedure is somewhat more complicated by the need to account for the jet curvature and unequal entrainment on either side of the jet. Accordingly, the viscous-inviscid matching procedure contains an additional loop for converging jet trajectories. The exit pressure matching loop is unchanged.

6.3.1 Viscous-Inviscid Matching

Unlike the single-jet ejector, the jets in the dual-jet ejector are not issued along the channel symmetry plane. With no geometric symmetry imposed at their centerlines, the jets have the freedom do develop asymmetric characteristics. Both curvature of the jet centerline and unequal entrainment on the two sides of the jet are additional effects which the viscous-inviscid interaction scheme for the dual-jet ejector must account for.

While the flow is not expected to be symmetric with respect to the individual jet centers, the overall flow is still symmetric with respect to the channel symmetry plane. As in the single-jet case, it is again sufficient to consider only the upper half of the ejector channel. The upper half-plane now contains one entire jet as opposed to the half jet encountered previously. Both the upper and lower surfaces of this jet must be treated separately, since in the absence of symmetry, the viscous-inviscid interaction taking place at the upper surface is different from the interaction taking place at the lower surface. Accordingly, the iteration scheme is extended accordingly to separately match the viscous and inviscid solutions at upper and lower interfaces.

The procedure for matching both sides of the jet is patterned after the one-sided matching done in the single-jet case. The suction velocities for the panels covering both sides of the jet are initially set to reflect an initial guess for the jet entrainment. The suction velocities applied to both the lower and upper control stations are set to the same fixed value. The inviscid problem is then solved, and the velocity components as well as the pressure at both the upper and lower side of the jet boundary calculated. Next the quantities $\dot{u}_0$, $\dot{a}$, and $\dot{\bar{p}}$ are determined. These terms are then used as boundary conditions in the solution of the viscous region (see Eq. (5.68)). Once the viscous solution is complete, the entrainment velocity at both the upper and lower interfaces are calculated. Finally, corrections to the panel suction velocities on both the upper and lower sides of the jet are made with the same iteration scheme given in Eq. (6.1). The cycle is repeated until the flow variables are continuous at both the upper and lower viscous-inviscid interfaces.

6.3.2 Exit Pressure Matching

The exit pressure matching procedure for the dual-jet ejector is exactly the same as that used in the single-jet case.

6.3.3 Determination of the Jet Trajectory

A new procedure needs to be introduced to account for the jet curvature. As discussed in Section 5.7.1, the jets curve in response to the pressure difference acting across them. The actual path of the jets is not known a priori, but rather must be determined as part of the solution. This requires an additional iteration loop to be built around the exit pressure matching and viscous-inviscid matching loops.

The viscous-inviscid interfaces are curved in proportion to the curvature of the jet centerline. The inviscid solution must account for this by distributing the panels which cover the jet boundary over an appropriately curved surface. The shape of this surface is not known ahead of time since it is dependent on the yet unknown distribution of pressure within the ejector inlet. The panels that form the viscous-inviscid interfaces must be free to move during the iteration process so that the jet trajectory remains compatible with the rest of the solution. The procedure used here is to initially guess the jet trajectory. The panels are laid out accordingly and both the viscous-inviscid and exit pressure matchings done. When the provisional solution is complete, the computed jet trajectory is compared with the initial guess. If the vertical distance between the two exceeds a specified tolerance at any point, the newly computed trajectory is used as the initial guess for the next iteration.

This process converges rapidly. It is a rare case where more than two cycles are needed to converge the jet trajectory. A summary of the overall iteration strategy for the dual-jet ejector is shown in Figure 6.4.

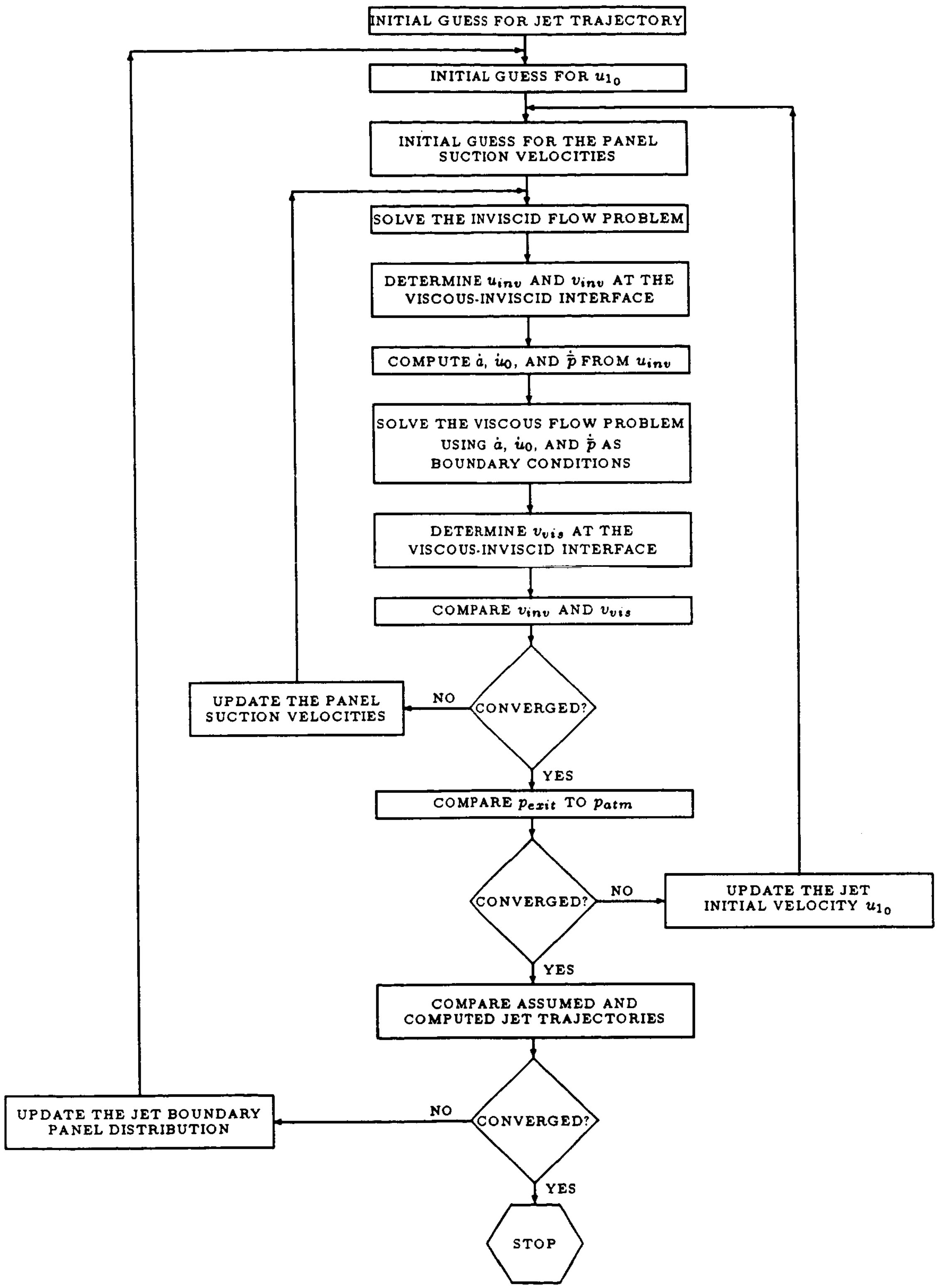


Figure 6.4: Iteration scheme for the dual-jet ejector. Note three levels of nesting.

Chapter 7

Results

In this chapter the predictions of the viscous-inviscid ejector algorithm are carefully examined. In an effort to validate the computer code, the results predicted for the single-jet ejector are compared against experimental data. The computations are then extended to an investigation of the effect of ejector geometry on performance. This is done for both the single-jet and dual-jet ejectors by systematically varying the primary nozzle position, ejector length, free stream speed, diffuser angle, and diffuser slope. The results of the parametric studies for the single-jet ejector are compared with experimental data for qualitative agreement. Finally, the computer code is used as a subroutine to an optimization package to demonstrate the suitability of the algorithm to practical design problems.

In all cases the results are presented in non-dimensional form where the thrust augmentation ratio, defined as

$$\phi = \frac{\text{Total Ejector Thrust}}{\text{Thrust of an Identical Nozzle Issued in Isolation}} \tag{7.1}$$

is plotted against non-dimensional forms of the individual parameters.

7.1 Comparison With Experiment

The predictions of the viscous-inviscid algorithm are compared with a series of measurements taken by Bernal and Sarohia[47] at the Jet Propulsion Laboratories

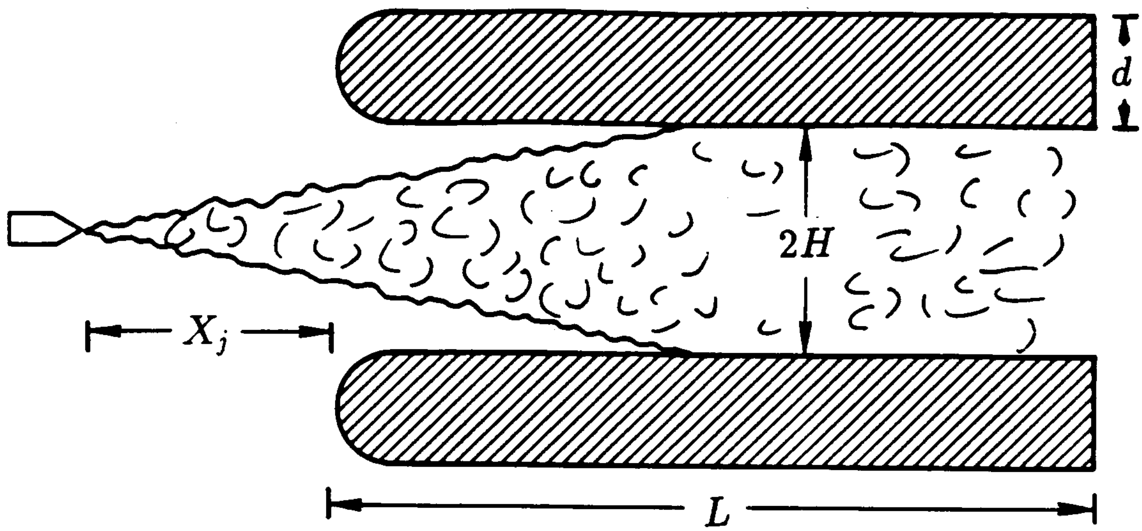


Figure 7.1: Experimental configuration. $L/2H = 3.25$, $x_j/2H = 1.0$, $d/2H = 0.5$, $U_\infty = 0.0$

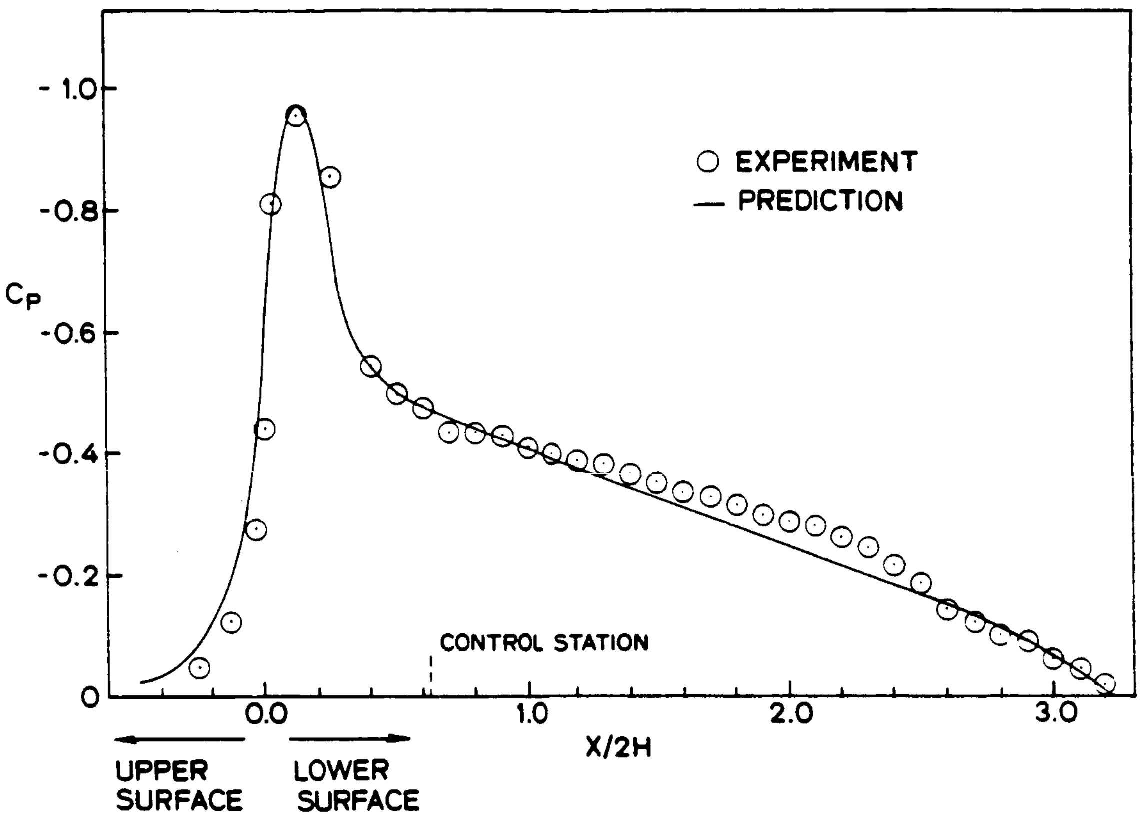
in 1982. Figure 7.1 shows a cross-section of the two-dimensional test configuration. The ejector shroud is composed of two thick flat plates with semi-circular leading edges. The plates are spaced so that the length to width ratio of the mixing chamber is 3.25. The primary nozzle is displaced one channel-width in front of the ejector. The jet exit Mach number is 0.3 in the experiment and no free stream is present.

7.1.1 Surface Pressure

Figure 7.2 shows a comparison between the measured and computed distribution of the ejector surface pressure. The results are presented in non-dimensional form where the surface pressure coefficient, defined as

$$C_p = \frac{p - p_\infty}{T_0/2H} \tag{7.2}$$

is plotted against the normalized surface coordinate. The viscous-inviscid algorithm does a good job at capturing the suction peak resulting from the acceleration of the



$$C_p = \frac{p-p_\infty}{T_0/2H}$$

Figure 7.2: Surface Pressure Comparison. $C_p = \frac{p-p_\infty}{T_0/2H}$

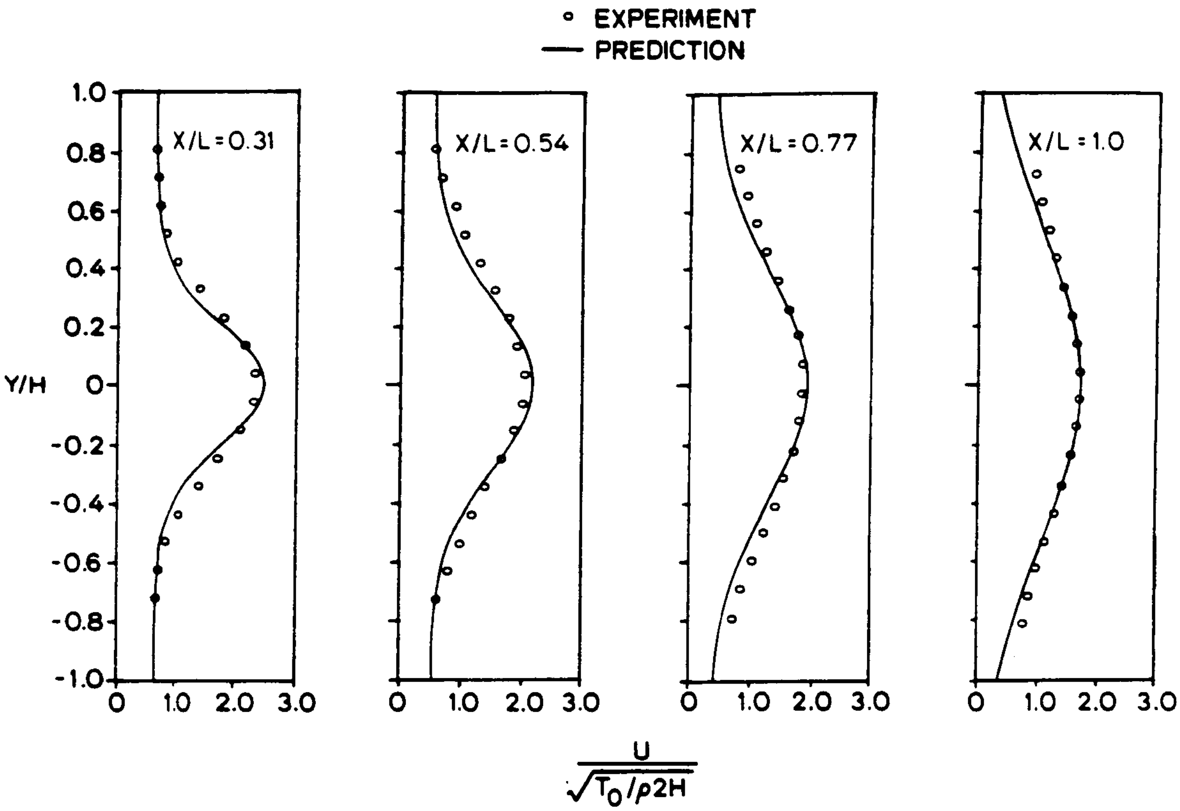


Figure 7.3: Comparison of the jet velocity profiles

secondary fluid as it flows around the shroud leading edge. The computed results also accurately predict the pressure recovery that results from the dissipation of momentum within the mixing region of the ejector channel. The fact that the conversion of the primary jet momentum to pressure is accurately predicted suggests that the simple algebraic turbulence model is doing an adequate job of simulating the turbulent shearing stresses.

7.1.2 Velocity Profile Evolution

Shown in Figure 7.3 is a comparison of the computed and measured velocity profiles within the ejector channel. The viscous-inviscid algorithm accurately predicts the jet spreading as well as the decay of the maximum velocity. The correct prediction

THRUST AUGMENTATION RATIO	
EXPERIMENT	$\phi = 1.2$
COMPUTATION	$\phi = 1.26$

Table 7.1: Thrust augmentation ratio comparison

of the jet growth provides additional justification for the use the algebraic turbulence model. Agreement in the shape of the velocity profiles demonstrates that the gaussian exponential velocity profile shape chosen for the viscous calculation is a good choice for representing the physics of the single-jet ejector flow.

7.1.3 Thrust Augmentation Ratio

The computed value of the thrust augmentation ratio is compared with the experimental value in Table 7.1. The good agreement demonstrates that the viscous-inviscid algorithm accurately models the overall ejector mixing process. The fact that the computed result is 5 percent higher than the experimental result could be attributed to the lack of account for skin friction in the computation.

Now with the results of the computation validated against experimental data, a series of parametric and optimizations are performed.

7.2 Parametric Studies

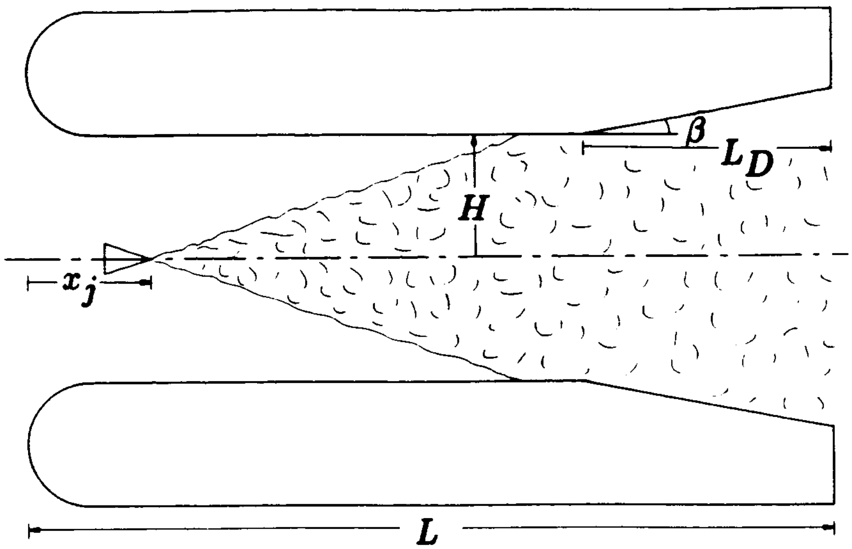
The effects of varying several geometrical parameters is investigated by perturbing the configuration that was used for the comparison with experiment. Unfortunately the experimental tests at JPL did not include any such geometrical parametric variations. Other experimentalists[1] have published data showing the effects of variation in one or two geometrical parameters. Outside of these limited results, there does not seem to exist a cohesive set of experimental data where a single configuration is subjected to systematic variations in several different geometric parameters. For this reason it is difficult to make a direct comparison of the computed results with experimental data when a large number of parameters are systematically varied. It is possible, however, to make a qualitative comparison with the available experimental data. The experimental data must be drawn from several independent tests that involve different basic geometrical configurations. No attempt is made to tailor the computational geometry to match each of these individual tests, but rather the basic computational geometry is held fixed and comparisons are made to show similar trends as opposed to exact agreement. Experimental data is more abundant for single-jet configurations and thus comparisons are made for this case only.

7.2.1 Parameters Varied

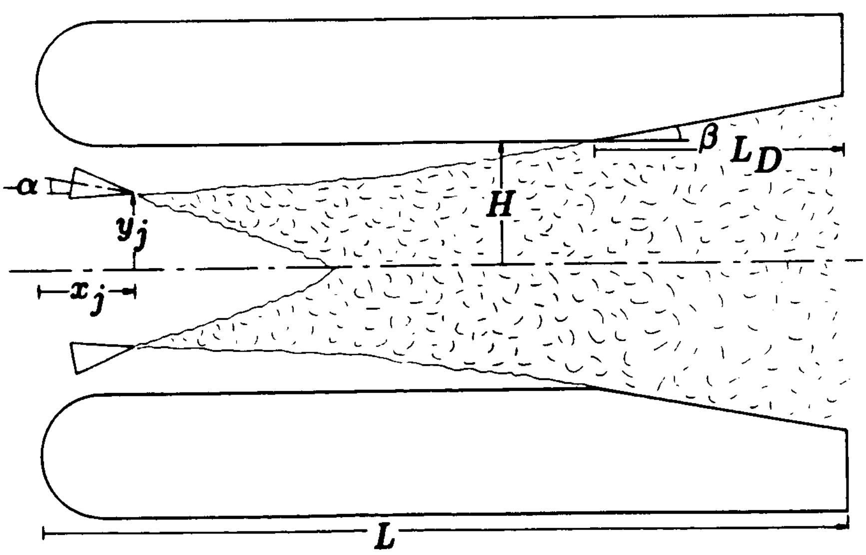
Figure 7.4 shows the geometrical parameters that are varied for both the single-jet and dual-jet ejectors. In non-dimensional form the parameters are: longitudinal nozzle placement, $x_j/2H$, ejector length, $L/2H$, free stream speed, $\gamma^2 = \rho U_\infty^2 H/T_0$, diffuser length, L_D/L, and diffuser angle β. The lateral nozzle placement, y_j/H and the nozzle tilt, α are additional parameters for the dual-jet ejector. Each of these parameters is varied independently while all others are held fixed at their nominal values, $x_j/2H = 0$, $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0$, and $\beta = 0$ for the single-jet ejector, and $x_j/2H = 0.44$, $y_j/H = 0.5$, $\alpha = 0$, $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0$, and $\beta = 0$ for the dual-jet configuration. For the single-jet case, the basic configuration is the same as the JPL test with the exception that the nozzle is located at the entrance plane of the ejector as opposed to one channel width in front. The nominal dual-jet configuration is the same as the single-jet one with the primary jet divided in two symmetrically placed jets of half the single jet intensity.

7.2.2 Ejector Length

Figure 7.5 shows the variation in the thrust augmentation ratio with ejector length for the single-jet and dual-jet configurations. The computations show that the ejector performance increases monotonically with the ejector length. This result can be explained as follows. As the ejector becomes longer, the high energy jet fluid

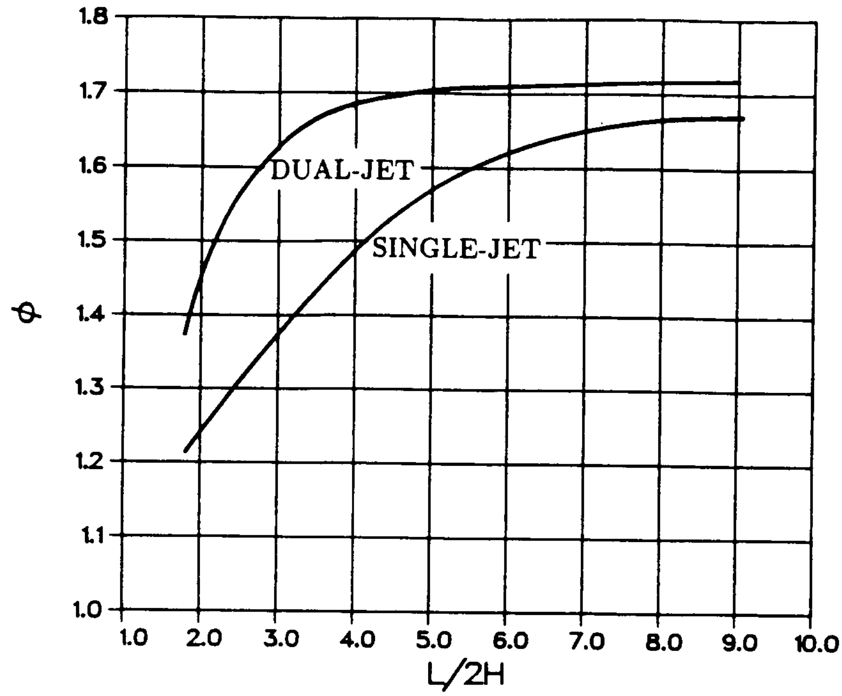


SINGLE-JET

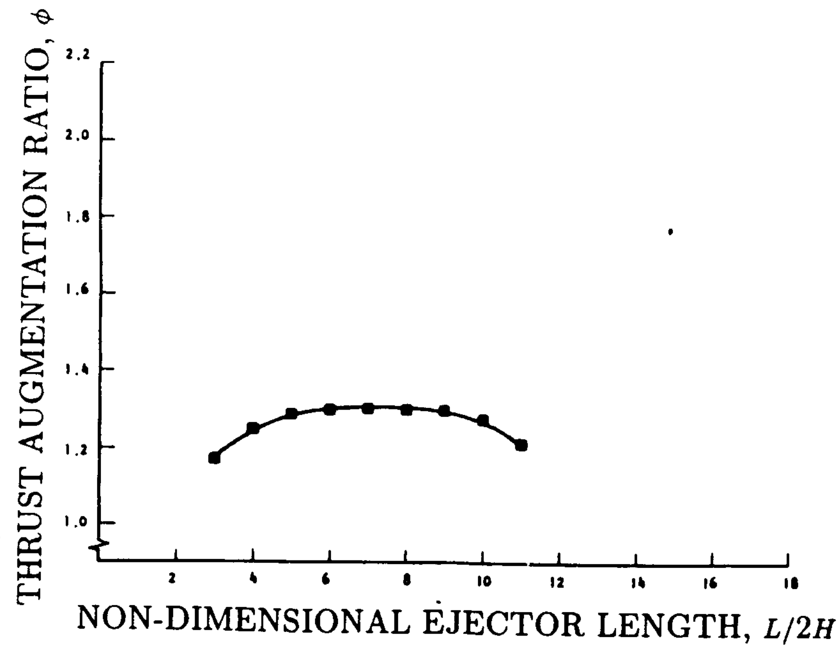


DUAL-JET

Figure 7.4: Variation of ejector geometrical parameters



(A)



(B)

Figure 7.5: Effects of the ejector length. (A) Computed results for single-jet: $x_j/2H = 0$, $\gamma = 0$, $L_D/L = 0$, $\beta = 0$, dual-jet: $x_j/2H = 0.44$, $y_j/H = 0.5$, $\alpha = 0$, $\gamma = 0$, $L_D/L = 0$, $\beta = 0$. (B) Qualitative comparison with experiment for a single-jet configuration (taken from Ref. [1]).

has more opportunity to mix with the ambient fluid. The enhanced mixing requires an increase in the amount of entrained secondary flow and hence an increase in performance.

The experimental results shown in part (B) of Figure 7.5 are for a single-jet configuration. The experimental data shows a similar trend with the exception that the thrust augmentation ratio does not increase monotonically with the ejector length. The experimental data show an increase in performance up to a maximum value at roughly $L/2H \simeq 7$, after which the performance continually degrades. The differences between the computed and experimental results arises from the neglect of skin friction in the computation. In the experiment, the increment in drag due to skin friction starts to overcome the increment in performance due to increasing the ejector length at $L/2H \simeq 7$. Without account for the viscous drag, the computed results are unable to show the optimal ejector length.

A very simple analysis can be made to better understand the differences in performance of the single and dual-jet ejectors. The premise of this analysis is that the dual-jet ejector has a greater effective length than does the single jet configuration. The basic idea is shown in Figure 7.6. By virtue of symmetry at the ejector channel centerline, each jet in the dual-jet configuration acts as a separate ejector. The length to width ratio of the two effective ejectors, however, is not the same as the original $L/2H$. The overall ejector length L is the same, but the effective channel width is reduced. If the nozzle is located midway between the ejector centerline and the ejector wall, then the effective channel width is $H/2$. In general, the effective channel width depends on the lateral position of the primary nozzle. For an arbitrary lateral nozzle position, the following hypothesis for the effective channel width is used

$$H_{eff} = \begin{cases} \left(1 - \frac{Y_j}{H}\right) H & \frac{Y_j}{H} < \frac{1}{2} \\ \left(\frac{Y_j}{H}\right) H & \frac{Y_j}{H} > \frac{1}{2} \end{cases} \tag{7.3}$$

The effective ejector length to width ratio is then

$$\left(\frac{L}{2H}\right)_{eff} = \frac{H}{H_{eff}} \left(\frac{L}{2H}\right) \tag{7.4}$$

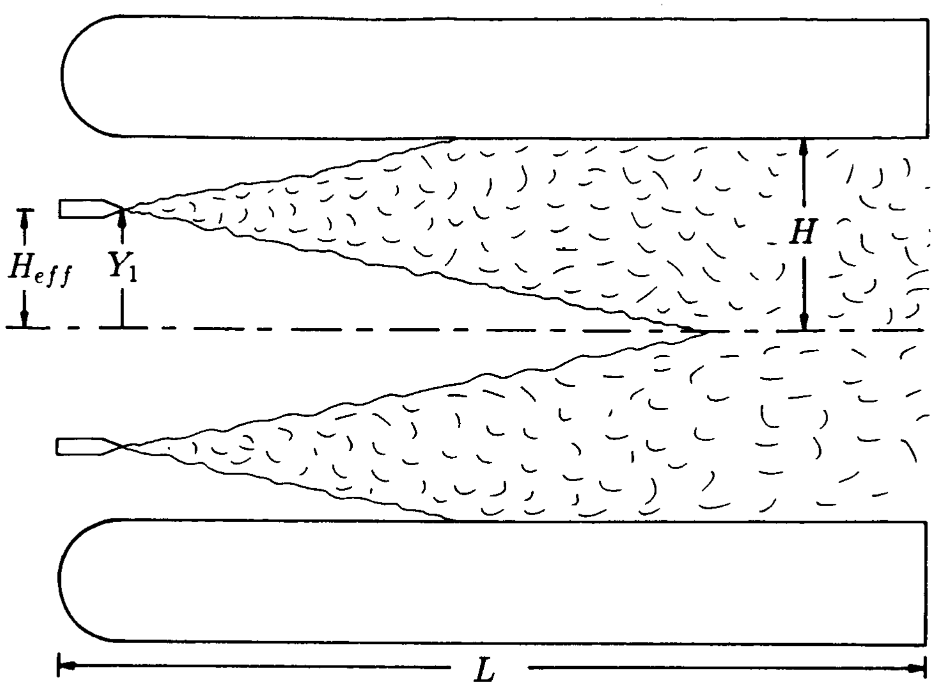


Figure 7.6: Effective ejector length for the dual-jet configuration

The thrust augmentation of the dual-jet ejector may then be related to the thrust augmentation ratio of an effective single-jet ejector as follows

$$\phi_{dual-jet}\left(\frac{L}{2H}, \frac{Y_j}{H}\right) \simeq \begin{cases} \phi_{single-jet}\left(\frac{1}{1-\frac{Y_j}{H}}\left(\frac{L}{2H}\right)\right) & \frac{Y_j}{H} < \frac{1}{2} \\ \phi_{single-jet}\left(\frac{1}{\frac{Y_j}{H}}\left(\frac{L}{2H}\right)\right) & \frac{Y_j}{H} > \frac{1}{2} \end{cases} \tag{7.5}$$

For the symmetrical placement $Y_j/H = 0.5$, the above relation becomes

$$\phi_{dual-jet}\left(\frac{L}{2H}, 0.5\right) \simeq \phi_{single-jet}\left(2\left(\frac{L}{2H}\right)\right) \tag{7.6}$$

Thus the performance of a dual-jet ejector with the nozzles symmetrically placed is predicted to perform the same as a single-jet ejector of twice the length. The validity of this estimate is demonstrated in Figure 7.7 where the results of the single-jet computation are used to provide an estimate of the dual-jet performance. The estimate agrees well with the actual dual-jet computation over the entire range of ejector lengths. An estimate for the variation in performance with the lateral position of the nozzles in the dual-jet ejector can also be made. Figure 7.8 shows a

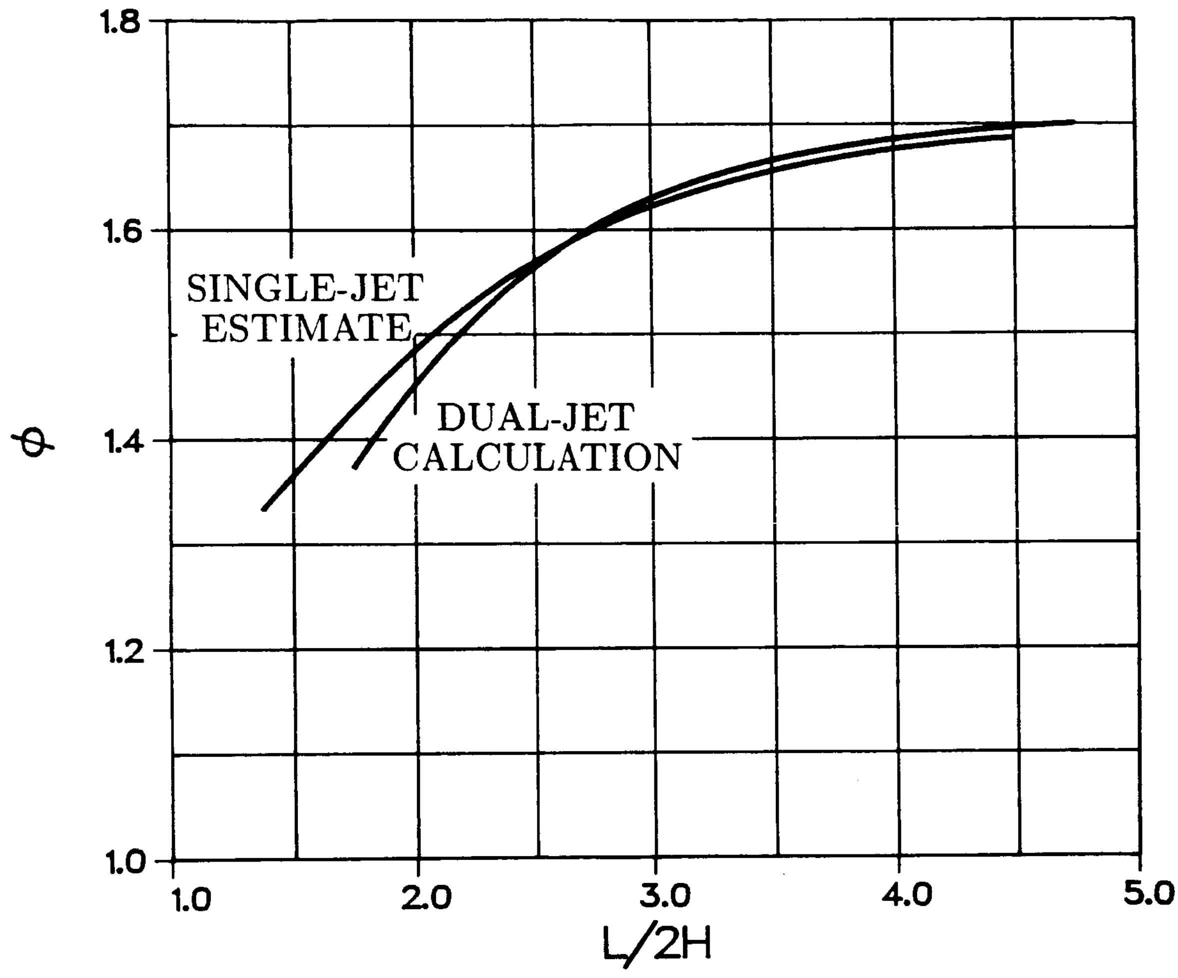


Figure 7.7: Comparison of the computed results for the dual-jet ejector with an estimate based on the effective single-jet ejector concept

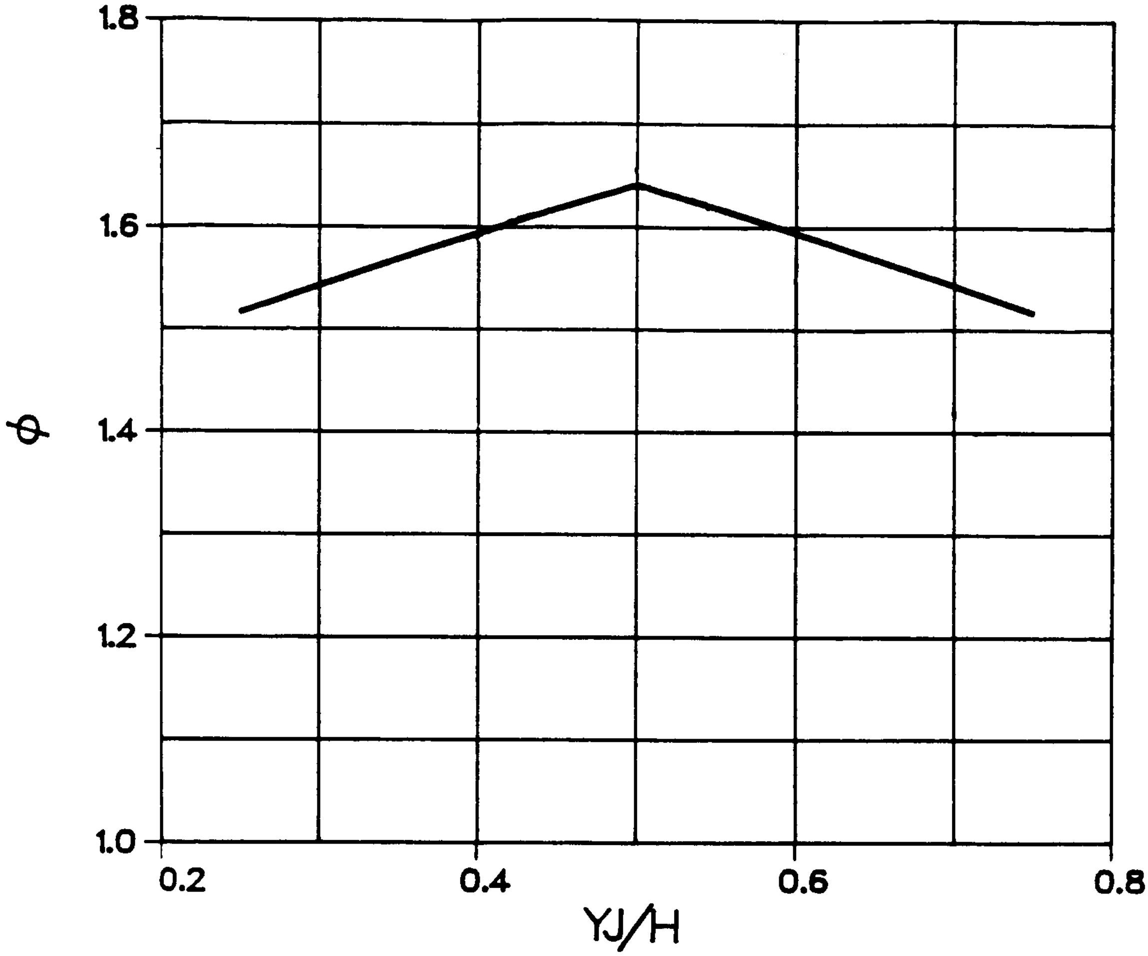


Figure 7.8: An estimate of the effect of the lateral position of the primary nozzles

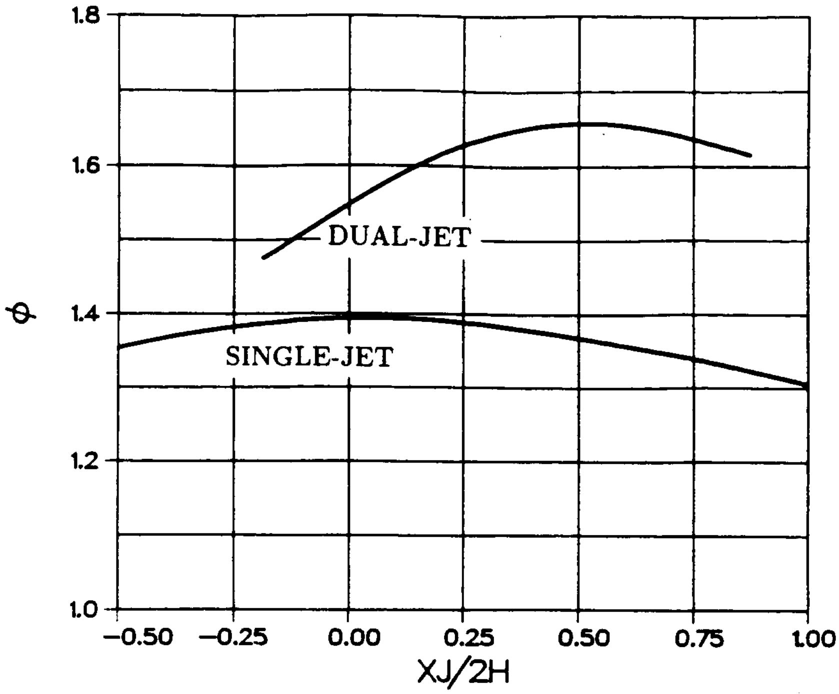
plot of Eq. (7.5) where the results for the single-jet computation have been used. The analysis shows that the optimal location for the nozzles is midway between the ejector channel centerline and the channel wall.

7.2.3 Longitudinal Nozzle Position

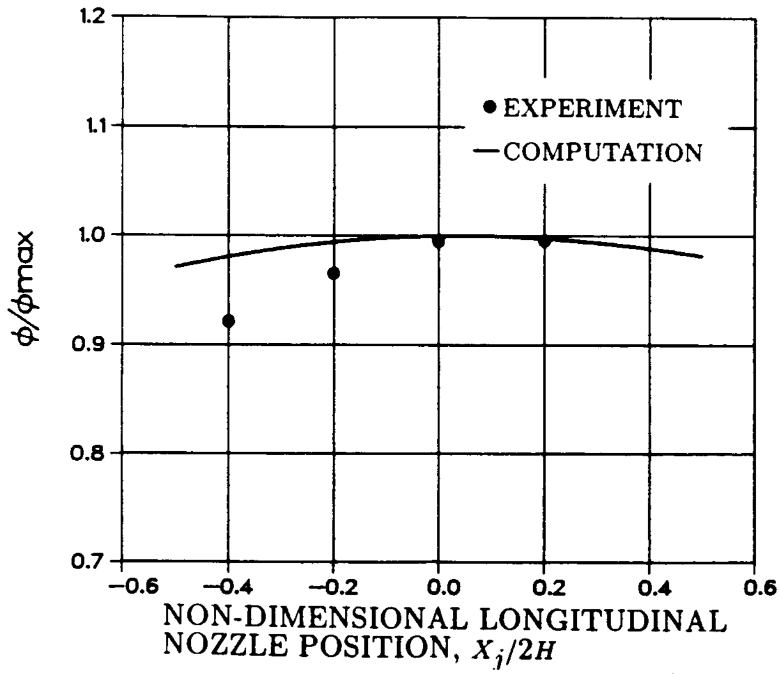
Shown in Figure 7.9 is the effect of the longitudinal nozzle placement for both the single-jet and dual-jet ejectors. The computed results for the single-jet ejector show that the performance is maximized when the nozzle is located at the ejector inlet plane. This fact may be explained as follows. When the nozzle is located ahead of the ejector, much of the entrainment takes place ahead of the ejector as well. The jet is already partially mixed as in enters the ejector shroud and as a result has less available kinetic energy to be used to entrain additional ambient fluid. The momentum flux of the secondary fluid entering the ejector is consequently reduced and the performance is degraded. As the nozzle is moved away from the entrance plane into the channel, the length over which the turbulent mixing can take place is reduced. A reduction in mixing again implies a reduction in the entrainment of ambient fluid and a corresponding drop in performance. According to this argument, the optimal nozzle location should be at the ejector entrance plane since at this location the jet has the greatest available kinetic energy as well as the longest distance within the channel for the mixing to take place.

In part (B) of Figure 7.9, a qualitative comparison with experimental data for a single-jet ejector is made. The experimental results show the same trend where the performance is maximized near the ejector entrance plane. The experiment shows a more rapid decrease in performance as the nozzle is moved in front of the ejector. This discrepancy is probably due to differences in the basic geometry of the experimental and computational configurations.

The computed results for the dual-jet ejector are similar to the single jet case with the exception that the maximum performance is obtained when the jets are located slightly inside the ejector channel. The fact that the optimal position is not at the entrance plane for the dual-jet ejector is related to the curvature of the



(A)



(B)

Figure 7.9: Effects of the longitudinal nozzle position. (A) Computed results for single-jet: $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0$, $\beta = 0$, dual-jet: $L/2H = 3.25$, $y_j/H = 0.5$, $\alpha = 0$, $\gamma = 0$, $L_D/L = 0$, $\beta = 0$. (B) Qualitative comparison with experiment for the single-jet case (taken from Ref. [1]).

jet centerlines. Due to the non-uniform pressure in the inviscid field at the ejector inlet, the jets are induced to follow curved trajectories. The relative position of the jet centers are therefore displaced from their optimal position midway between the ejector centerline and the ejector wall. As the nozzles are moved further inside the channel, they are located in a region of increasing uniformity in the pressure field. Consequently the displacement of the jet centerlines diminishes and the performance is increased. The optimal position for the nozzles is the point where the rate of increase in performance due to a less deflected jet trajectory is equal to the rate of decrease in performance due to a decrease in the overall length over which the flow has to mix.

7.2.4 Lateral Nozzle Position for the Dual-Jet Ejector

Figure 7.10 shows how the lateral nozzle position affects the performance for the dual-jet ejector. The results show that the performance is maximized when the jets are located midway between the ejector walls and the channel centerline. The performance drops off a bit faster when the jets are moved towards the ejector channel walls than when they are moved towards the channel centerline.

A comparison of the the computed results with the estimate provided by the effective ejector length concept is shown in part (B) of figure 7.10. The qualitative agreement shows that moving the nozzle from its optimal location at the midpoint between the channel centerline and the channel wall results in a shortening of the effective ejector length.

7.2.5 Nozzle Tilt for the Dual-Jet Ejector

Shown in Figure 7.11 is the variation in performance of the dual-jet ejector with the primary nozzle tilt. The computation shows that, for the chosen position of the primary nozzle, the performance is maximized for a nozzle tilt of zero degrees. The nozzle tilt affects the performance in much the same way as does the nozzle lateral nozzle position since tilting the nozzle forces the jet centers to leave their optimal point midway between the ejector channel centerline and the ejector wall.

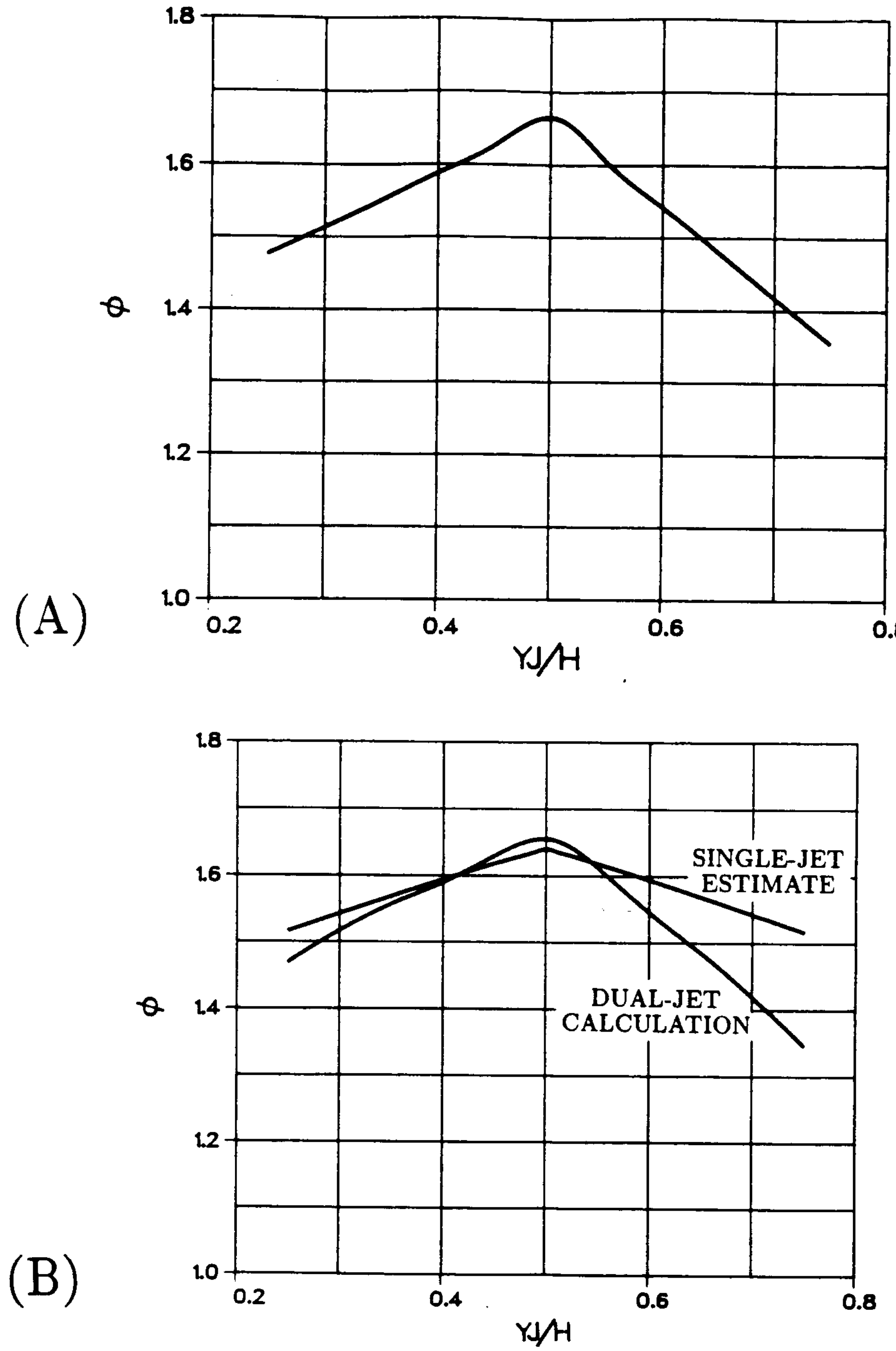


Figure 7.10: Effects of the lateral nozzle position for the dual-jet ejector. (A) Computation for $x_j/2H = 0.44$, $\alpha = 0$, $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0$, $\beta = 0$. (B) Comparison with the estimate based on the effective ejector concept.

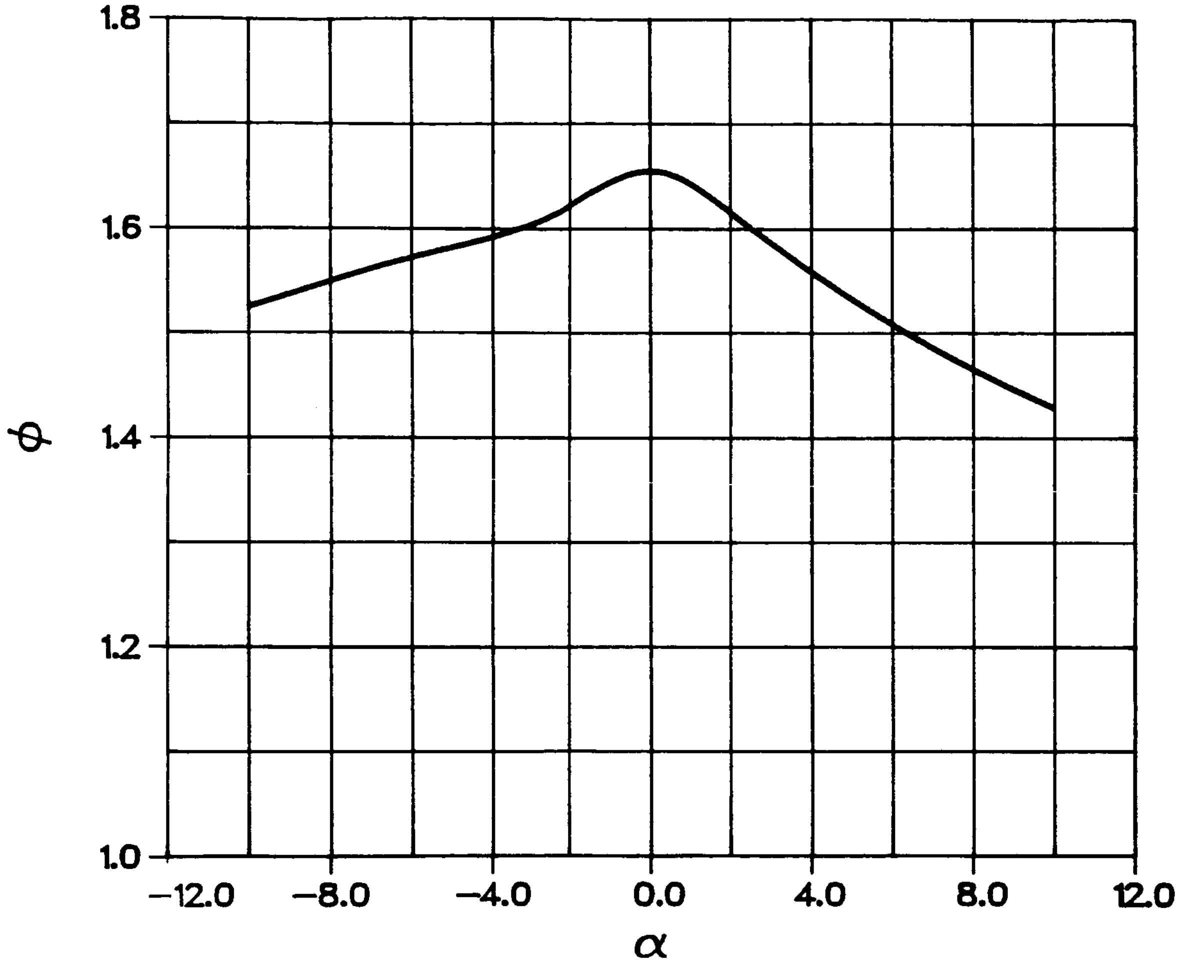


Figure 7.11: Effects of the nozzle tilt for the dual-jet ejector. $x_j/2H = 0.44$, $y_j/H = 0.5$, $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0$, $\beta = 0$.

7.2.6 Free Stream Speed

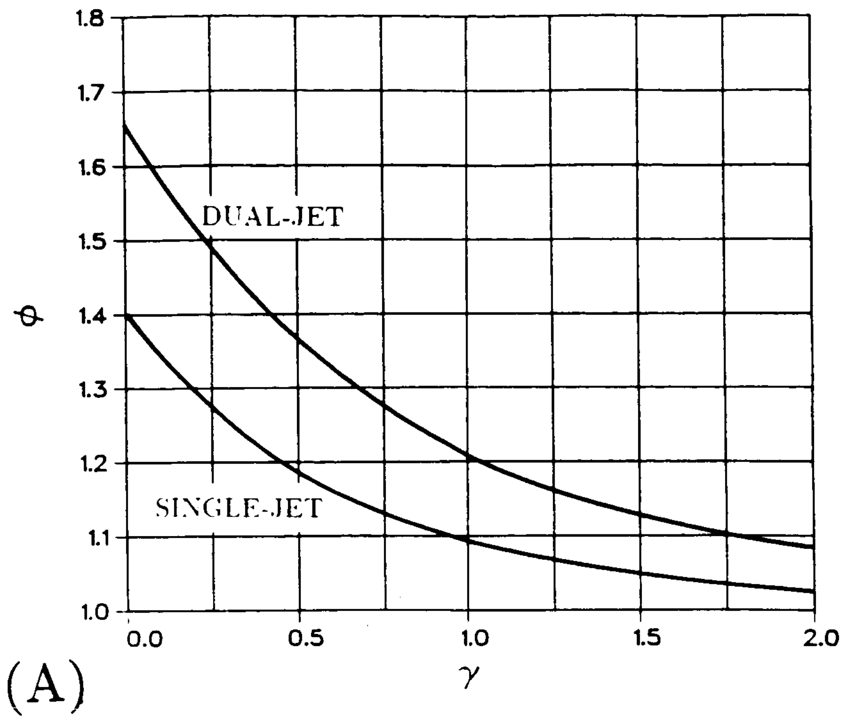
Figure 7.12 shows the computed variation in the thrust augmentation ratio with the free stream speed for the single-jet and dual-jet ejectors. The parameter γ is a non-dimensional measure of the free stream speed. Its square is proportional to the force created by the dynamic pressure of the free stream acting over the channel width, divided by the primary jet thrust.

$$\gamma^2 = \frac{\rho U_\infty^2 H}{T_0} \tag{7.7}$$

The computed results show a steady decrease in performance with increasing free stream intensity. The results for the single-jet and dual-jet ejectors show a similar trend, with the dual-jet ejector out-performing the single-jet ejector throughout the entire range of γ. The reason for the decrease in performance with increasing free stream speed is due to an increase in the ram drag. The experimental data for a single-jet configuration shown in part (B) of Figure 7.12 illustrates a similar trend. The results of the control volume analysis, shown in part (C) of Figure 7.12 again agree qualitatively with the results of the viscous-inviscid calculation. In comparing the viscous-inviscid results with the control volume analysis, it is evident that the dual-jet ejector achieves a higher degree of mixing (smaller exit velocity skewness factor, λ_2) than does the single-jet ejector. The higher degree of mixing enables the dual-jet ejector to maintain its advantage over the single-jet configuration as the free stream intensity is increased.

7.2.7 Diffuser Length

The computed variation in thrust augmentation ratio with diffuser length for a constant diffuser angle of 20^0 is shown in Figure 7.13. The computation shows that the thrust augmentation ratio is a non-monotonic function of the diffuser length when the diffuser angle is held fixed. For the single-jet case, the performance is maximized at about $L_D/L = 0.3$. The computation for the dual-jet ejector shows increasing performance over the entire range of diffuser lengths investigated. The



$$\gamma^2 = \frac{\rho u_\infty^2 H}{T_0}$$

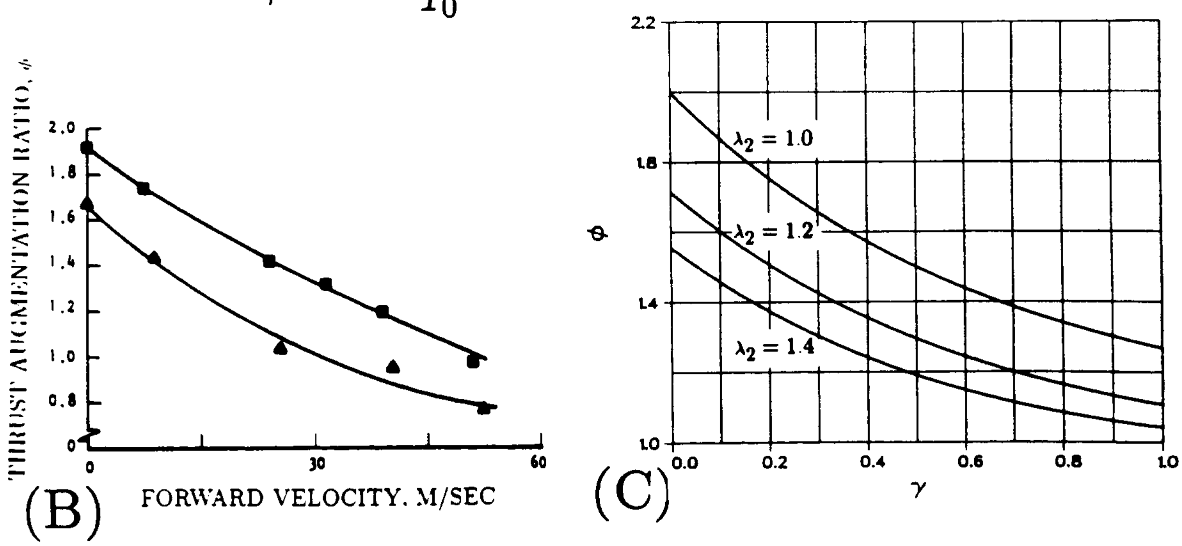


Figure 7.12: Effects of the free stream speed. (A) Computed results for single jet: $x_j/2H = 0$, $L/2H = 3.25$, $L_D/L = 0$, $\beta = 0$, dual-jet: $x_j/2H = 0.44$, $y_j/H = 0.5$ $\alpha = 0$, $L/2H = 3.25$, $L_D/L = 0$, $\beta = 0$. (B) Qualitative comparison with experiment for a single-jet configuration (taken from Ref. [1]). (C) Comparison with the control volume analysis.

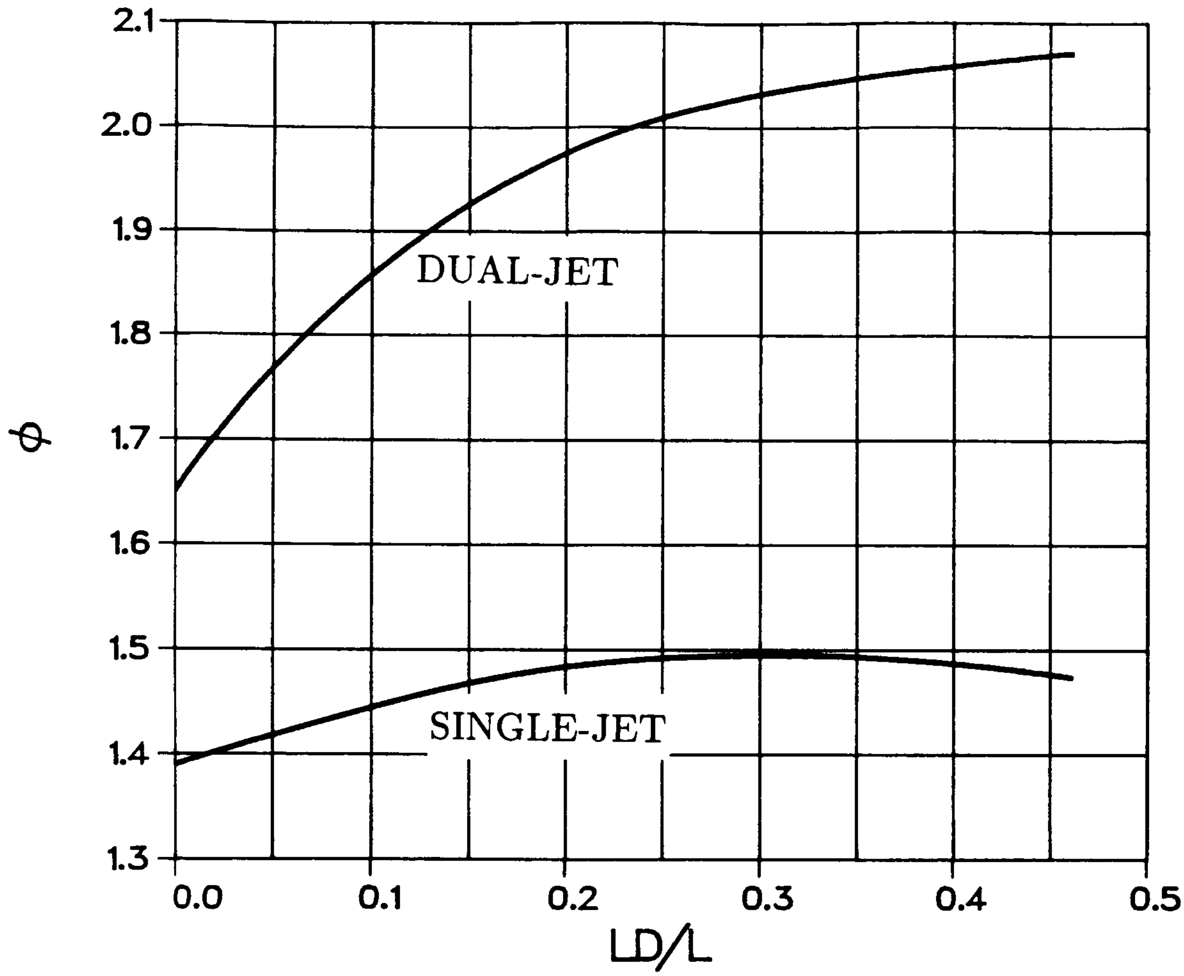


Figure 7.13: Effects of the diffuser length. Computed results for single-jet: $x_j/2H = 0$, $L/2H = 3.25$, $\gamma = 0$, $\beta = 20^0$, dual-jet: $x_j/2H = 0.44$, $y_j/H = 0.5$, $\alpha = 0$, $L/2H = 3.25$, $\gamma = 0$, $\beta = 20^0$

dual-jet performance should go through a maximum, however, but apparently at a value of L_D/L greater than 0.45.

A physical explanation of the effect of a diffuser is easier to understand if the results of Figure 7.13 are replotted as the thrust augmentation ratio versus the diffuser area ratio. Such a plot is shown in Figure 7.14. The results look much the same in this plot since the diffuser area ratio is directly proportional to the diffuser length if the diffuser angle is held fixed.

Recall that the ejector exit pressure must equal the atmospheric value. Then, if the turbulent mixing within the diffuser is neglected, the diffuser area ratio alone sets the pressure at the entrance of the diffuser to a value less than atmospheric. The lowered pressure within the ejector induces additional secondary flow to enter the device and hence an increase in performance. At the same time, however, the lowered pressure acting over the sloped diffuser walls creates a drag force. The drag force increases faster than does the increment in performance due to the the enhanced secondary flow. As the diffuser area ratio is increased, the pressure drag soon dominates and the thrust augmentation ratio falls from its maximum value.

The boundary layers within the diffuser are neglected in the viscous-inviscid algorithm and thus it is not possible to detect the decrease in performance associated with boundary layer separation from the diffuser walls when high area ratios are used. Thus the computed decrease in performance following the maximum value of thrust augmentation is due to increasing pressure drag and not diffuser stall. In interpreting the experimental data for the single-jet configuration shown for comparison in part (B) of Figure 7.14, it is difficult to determine whether the decrease in performance after the maximum value is due to boundary layer separation or from increasing pressure drag.

Part (C) of Figure 7.14 shows the corresponding result of the control volume analysis for comparison with the viscous-inviscid computation. The trends are seen to be quite similar.

In comparing the control volume results with the viscous-inviscid computation, it is again evident that the dual-jet ejector achieves a higher degree of mixing (lower exit velocity skewness, λ_2) than does the single-jet ejector. Because of its ability

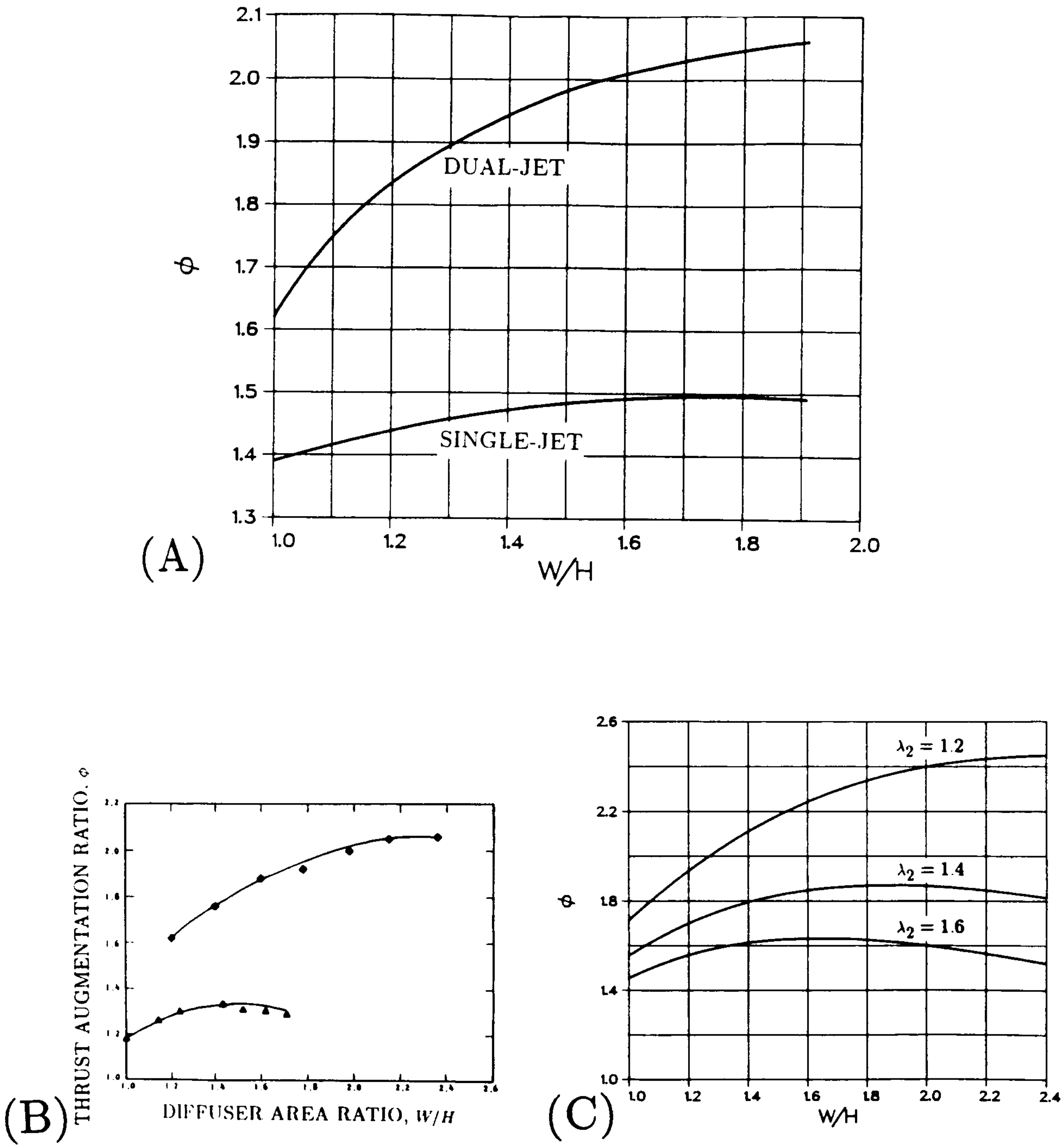


Figure 7.14: Effects of the diffuser area ratio (a replotting of Figure 7.13). (A) computed results for single jet: $x_j/2H = 0$, $L/2H = 3.25$, $\gamma = 0$, $\beta = 20^0$, dual jet: $x_j/2H = 0.44$, $y_j/H = 0.5$, $\alpha = 0$, $L/2H = 3.25$, $\gamma = 0$, $\beta = 20^0$. (B) Qualitative comparison with experiment for a single-jet configuration (taken from Ref. [1]). (C) Comparison with the control volume analysis.

to more efficiently mix the primary and secondary streams, the dual-jet ejector is predicted to perform significantly better than the single-jet counterpart when a diffuser is used. The advantage of the dual-jet ejector is most evident for greater diffuser area ratios.

7.2.8 Diffuser Angle

The variation in thrust augmentation ratio with diffuser angle for constant diffuser length is shown in Figure 7.15. The computed results look much the same as those for varying the diffuser length while holding the angle fixed (c.f. Figure 7.13). The similarity between the two sets of results suggests that the thrust augmentation ratio is predominantly a function of the diffuser area ratio and not the details of the diffuser shape. This hypothesis is tested by performing a computation where the diffuser length and diffuser angle are varied simultaneously in such a way that the diffuser area ratio remains fixed. The results of this computation are show in Figure 7.16. The flatness of the computed results indicates that the overall performance is nearly independent of the details of the diffuser shape. The code predicts only a slight advantage in using a short diffuser with a large angle. The fact that the thrust augmentation ratio is essentially independent of the diffuser shape indicates that there is a negligible amount of turbulent mixing taking place in the diffuser.

7.3 Optimization Studies

In an effort to demonstrate the usefulness of the viscous-inviscid algorithm for practical design problems, a few example optimization studies have been performed. In these studies, the viscous-inviscid computer code is used as a subroutine in an optimization package. Due to the efficient nature of the viscous-inviscid algorithm, configurations are optimized in manageable amounts of time on a VAX 11/780 machine.

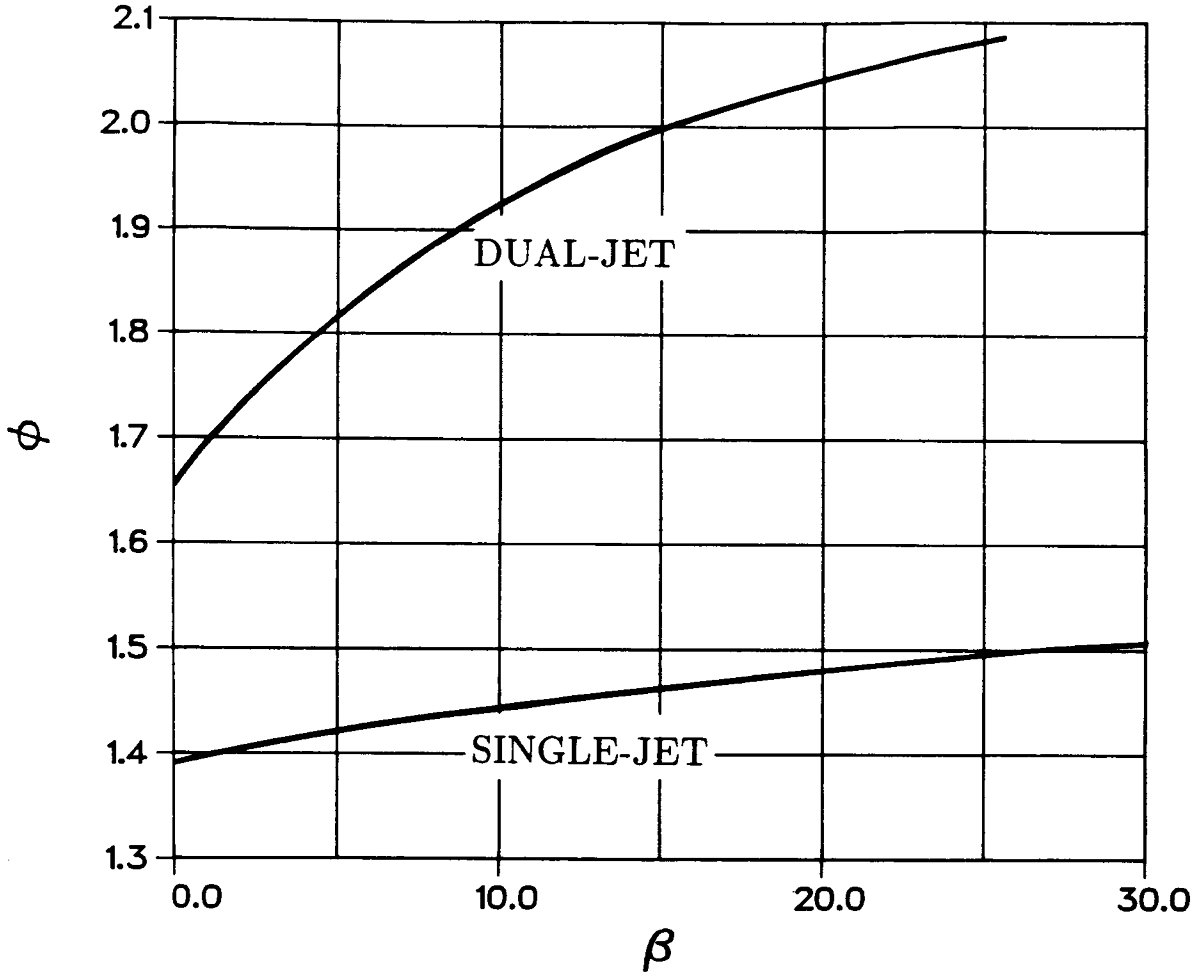


Figure 7.15: Effects of the diffuser angle for constant diffuser length. Computed results for single-jet: $x_j/2H = 0$, $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0.31$, dual-jet: $x_j/2H = 0.44$, $y_j/H = 0.5$, $\alpha = 0$, $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0.31$.

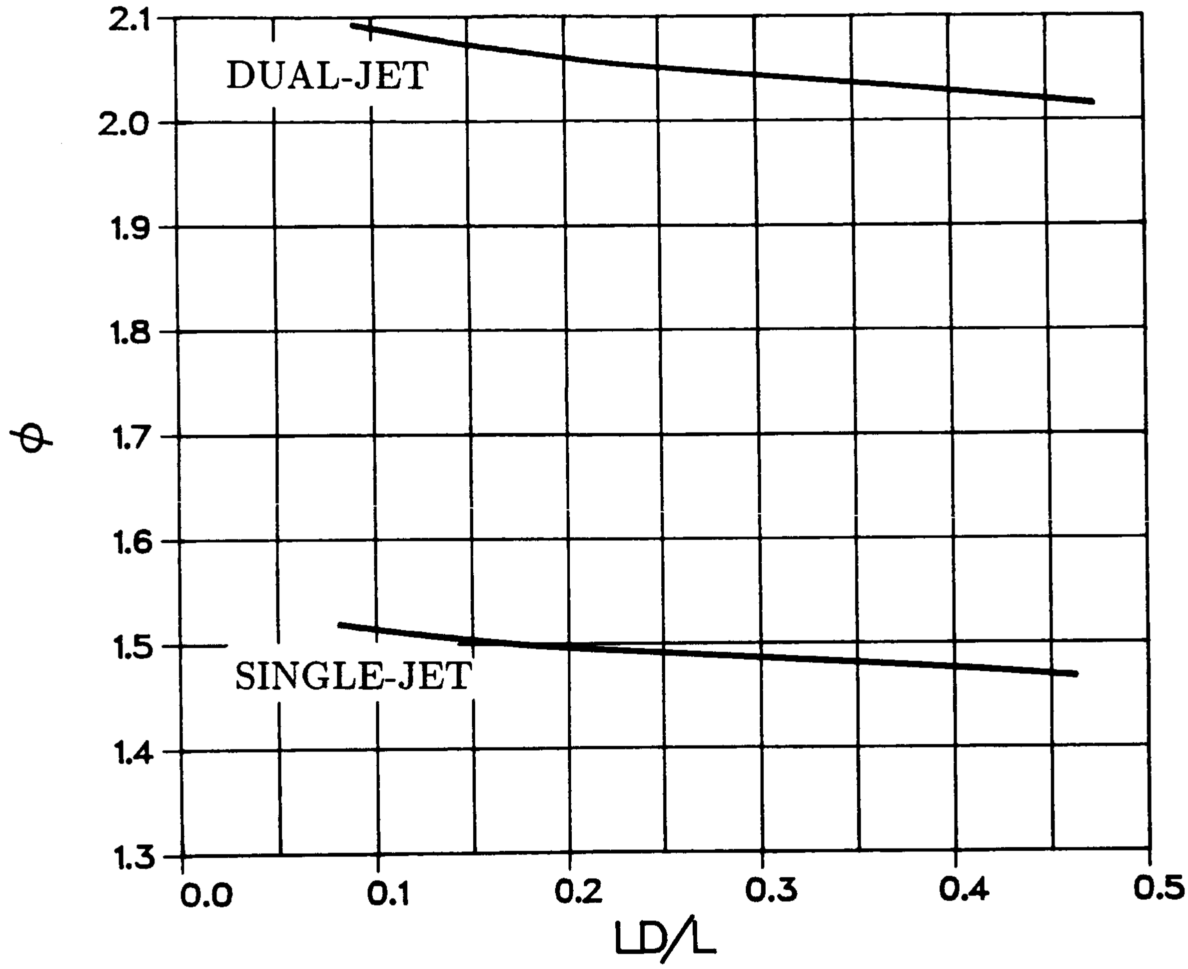


Figure 7.16: Effects of the diffuser length for constant diffuser area ratio. Computation for single-jet: $x_j/2H = 0$, $L/2H = 3.25$, $\gamma = 0$, $W/H = 1.73$, dual-jet: $x_j/2H = 0.33$, $y_j/H = 0.5$, $\alpha = 0$, $L/2H = 3.25$, $\gamma = 0$, $W/H = 1.73$.

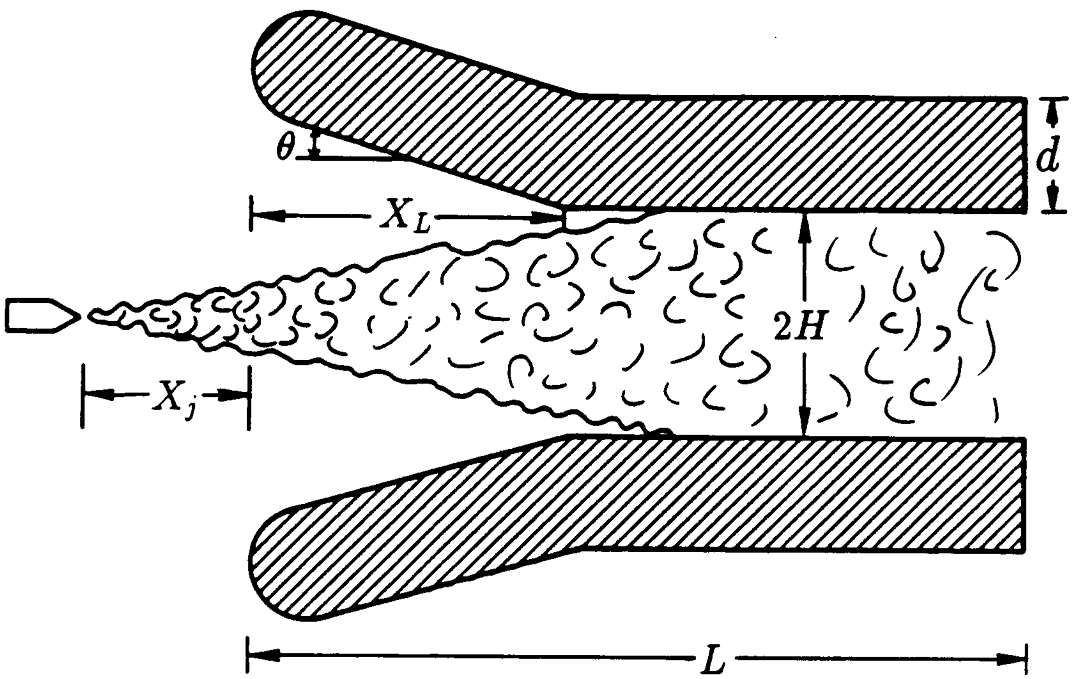


Figure 7.17: Configuration for the ejector inlet optimization. x_j, x_L, θ, U_∞, and the dynamic viscosity, μ are variable. Fixed parameters are: $L/2H = 3.25$, $d/2H = 0.5$.

7.3.1 Single-Jet Ejector Optimization

The computer code for the single-jet ejector is used to optimize a thrust augmentor inlet for several different flight conditions. The basic configuration is again the geometry used in the JPL test. Figure 7.17 shows the variable-geometry inlet to be used in the optimization study. The primary jet is free to move fore and aft of the ejector entrance plane. A variable-length section of the inlet is also free to rotate towards and away from the ejector centerline. In non-dimensional form the design variables are: nozzle position - $x_j/2H$, inlet lip length - $x_L/2H$, inlet lip rotation angle - θ, free stream speed - $\gamma^2 = \rho U_\infty^2 H/T_0$, and Reynolds number - $R_e = \sqrt{2H\rho T_0}/\mu$. The Reynolds number becomes an important parameter in the optimization study because a boundary layer calculation is included for the inlet portion of the ejector. Inlet geometries that result in boundary layer separation are rejected in the optimization process.

A quasi-Newton optimization package[65] is coupled with the viscous-inviscid code to systematically search through the design parameters. Constraints imposed by geometrical restrictions as well as boundary layer separation are incorporated into the optimization scheme through the use of algebraic penalty functions. The penalty functions artificially lower the performance once a constraint is violated.

The free stream speed parameter, γ and the Reynolds number R_e are chosen to define the flight condition. The optimization package then repeatedly evaluates the viscous-inviscid code to determine the optimal values of the remaining parameters. A concise statement of the optimization problem is

$$\text{MAXIMIZE} \quad \phi = \phi\left(\frac{x_j}{2H}, \frac{x_L}{2H}, \theta\right) \tag{7.8}$$

subject to the geometrical and boundary layer separation constraints.

Penalty Function Transformation

In its present form, the problem here is one of constrained optimization. Problems of constrained optimization are much more difficult to treat than are those of unconstrained optimization. Accordingly, a penalty function transformation[66] is used to transform the constrained optimization problem into one of unconstrained optimization. The idea behind the penalty functions is simple. The constraints are completely ignored until one of them is violated. When a constraint is violated, the performance is artificially lowered in an effort to redirect the search away from the forbidden region. The penalty functions thus simulate the effects of the constraints while allowing the problem to be treated under an unconstrained optimization framework.

With the use of penalty functions, the objective is to maximize the following

$$\text{MAXIMIZE} \quad g = \phi - \sum_{i=1}^{N} c_i \delta_i^2 \tag{7.9}$$

where the c_i are weighting factors and the δ_i are the penalty functions. The penalty functions themselves are composed of Heavyside functions. For example, a constraint of $\theta < \theta_0$ is modeled as

$$\delta = (\theta - \theta_0)\mathcal{H}(\theta - \theta_0) \tag{7.10}$$

where $\mathcal{H}$ is the Heavyside function. Note that the penalty is zero until $\theta = \theta_0$.

The weighting coefficients c_i are a measure of the relative importance of enforcing each constraint. Low values of c_i imply little attention paid to the constraints, while larger values increase their importance. The magnitude of the weights have a profound effect on the convergence of the optimization process. In general the convergence degrades with increasing values of the weights. The best strategy for obtaining convergence is to let the weights vary during the optimization process such that their magnitude is steadily increased as the optimal point is neared.

Optimal Solutions

Optimal configurations are determined for a wide range of Reynolds number for three values of the dimensionless free stream velocity, γ. Figure 7.18 shows the variation in the performance of a thrust augmentor with an optimized inlet as a function of both Reynolds number and free stream speed. The results indicate that the performance is an increasing function of Reynolds number, with strongest dependence in the low Reynolds number range. The rapid increase in performance at low Reynolds numbers is associated with transition from a laminar to a turbulent boundary layer. A laminar boundary layer can not withstand the severe adverse pressure gradient which is present in the inlet region. In an effort to avoid inlet stall, the optimization routine seeks a configuration that reduces the pressure rise in the inlet region by decreasing the degree of turbulent mixing within the shroud. In so doing, the performance is decreased since the mechanism of thrust augmentation relies on mixing of the high momentum jet with the ambient fluid. As the Reynolds number is increased to a value sufficient to induce transition to a turbulent boundary layer, the performance is greatly enhanced due to the fact that the turbulent boundary layer is able to negotiate the intensified pressure rise associated with increased mixing within the shroud.

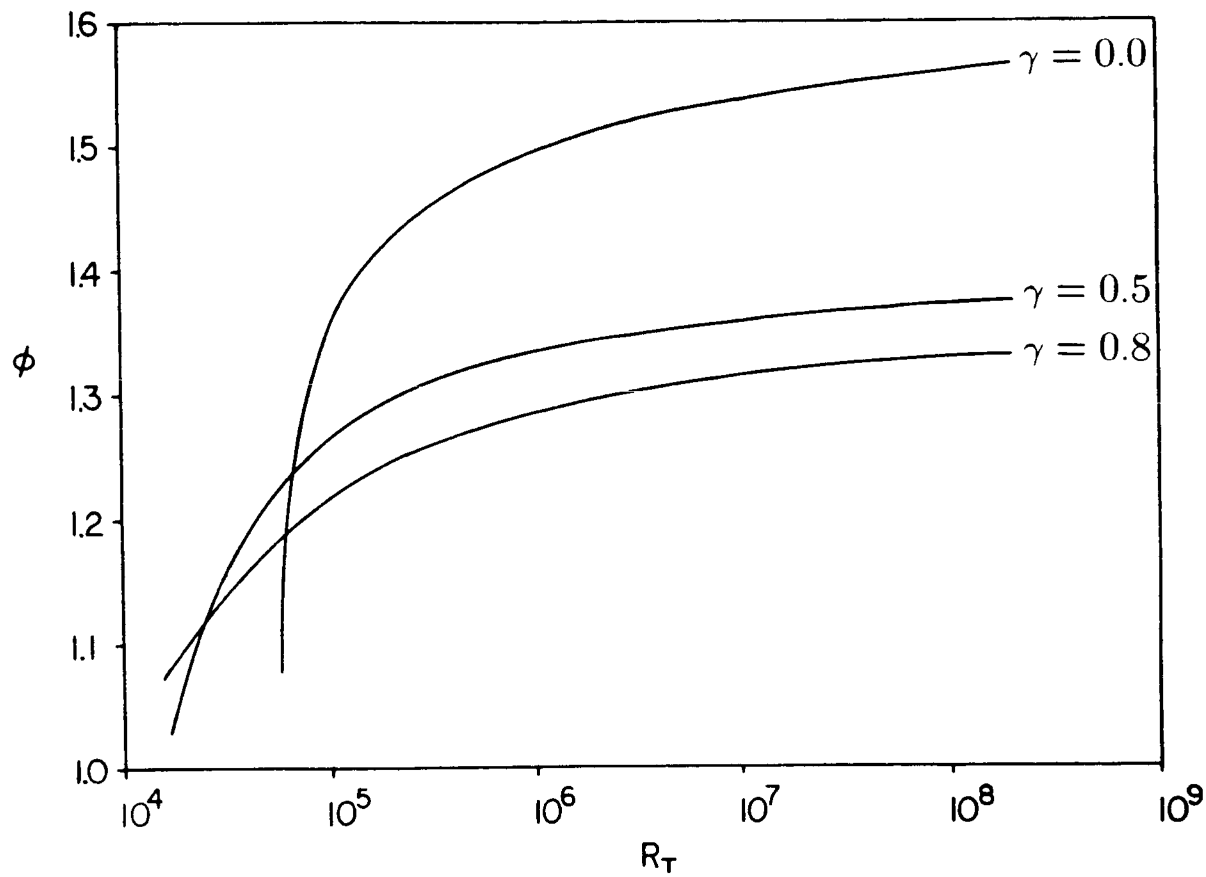


Figure 7.18: Performance of the thrust augmentor with an optimized inlet

When a non-zero free stream speed is included, the presence of a strong favorable pressure gradient following the stagnation point at the shroud nose helps to energize the boundary layer, thus making it more resilient to separation as the pressure rise in the inlet region is encountered. In contrast, for the case of static operation, the boundary layer begins at the tail end of the shroud, and due to its lengthy evolution and less favorable pressure gradient, becomes thick and sluggish by the time it has traveled the distance necessary to be swept into the inlet. The resulting thick, weak boundary layer experiences separation at a smaller pressure rise compared to the more favorably energized boundary layer. For this reason, increased levels of performance are noted in the laminar regime when a free stream velocity is present.

In the high Reynolds number regime, performance decreases with increasing free stream speed. This is due to an increase in the ram drag.

A few representative optimal shapes corresponding to the performance curves in Figure 7.18 are shown in Figures 7.19 and 7.20. The results show that the optimal design shapes are a much stronger function of Reynolds number than free stream speed. At low Reynolds number, Figure 7.19 shows that the optimal nozzle position is located up to one channel width ahead of the shroud, while the inlet is slightly expanded. This combination serves to minimize the adverse pressure gradient in the inlet region as required by the laminar boundary layer which develops there. In Figure 7.20 as the Reynolds number is increased and the boundary layers undergo transition, the nozzle moves approximately to the entrance plane of the shroud. The inlet lips rotate through the horizontal and then towards the jet as the Reynolds number is increased. The length of the inlet lip which is rotated is seen to increase with Reynolds number.

More detail on the behavior of the various design parameters as the Reynolds number and dimensionless free stream speed are varied is shown in the following sequence of plots. Figure 7.21 illustrates the optimal lip rotation angle as a function of Reynolds number for three values of the dimensionless free stream speed. It can be seen that the optimal lip rotation angles follow a similar trend for all three values of dimensionless free stream velocity. As the Reynolds number is increased, and laminar boundary layers undergo transition to turbulence, the lips rotate quickly

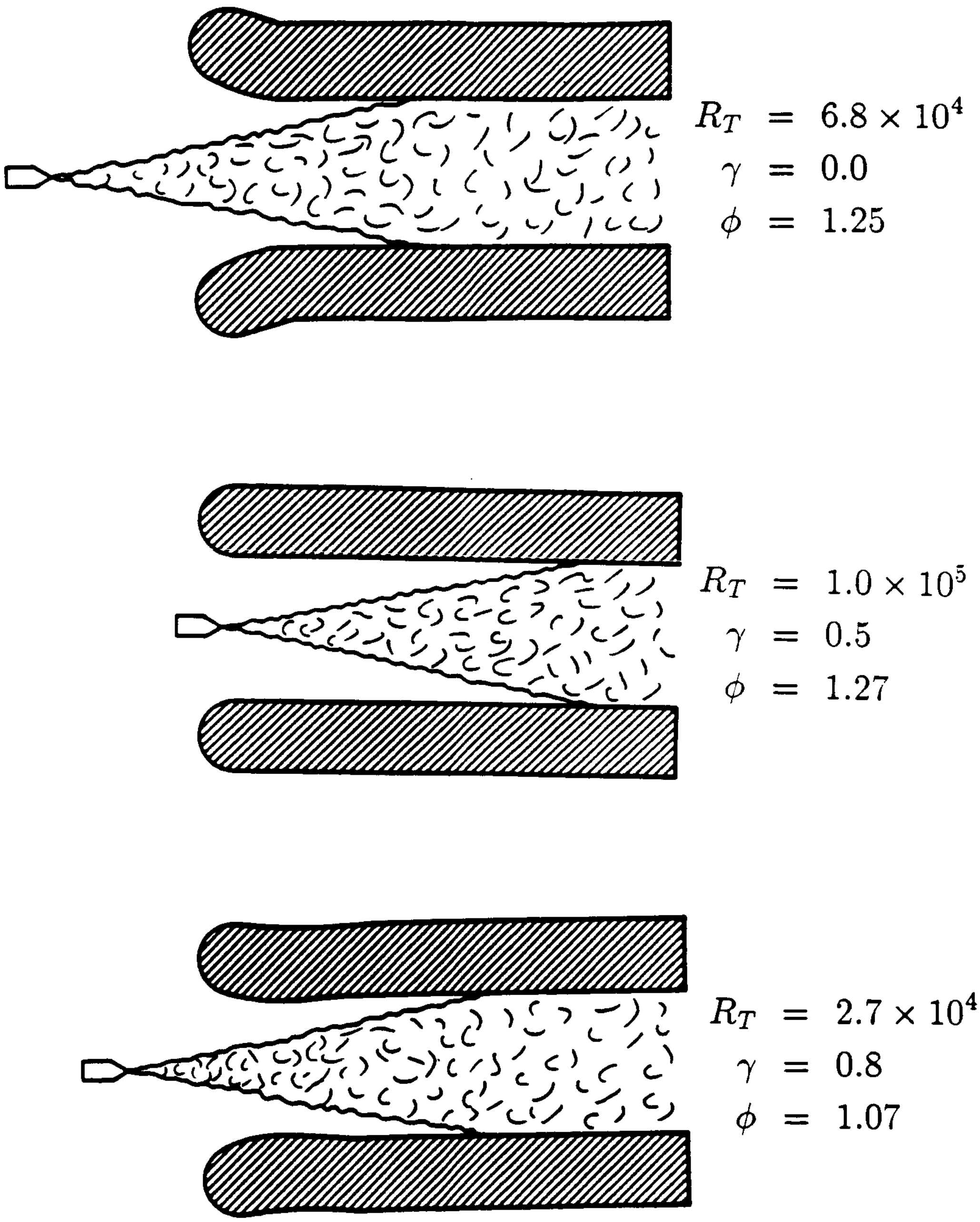
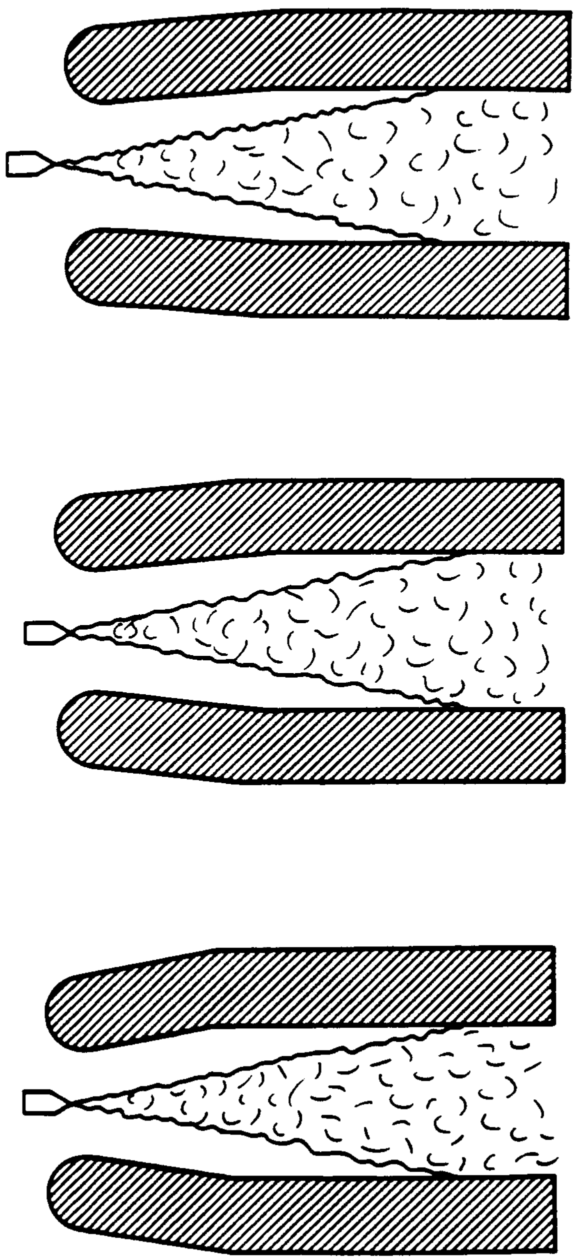


Figure 7.19: Optimal configurations at low and moderate Reynolds numbers



$$R_T = 1.0 \times 10^7$$
$$\gamma = 0.0$$
$$\phi = 1.54$$

$$R_T = 6.4 \times 10^6$$
$$\gamma = 0.5$$
$$\phi = 1.37$$

$$R_T = 9.2 \times 10^7$$
$$\gamma = 0.8$$
$$\phi = 1.31$$

Figure 7.20: Optimal configurations at high Reynolds numbers

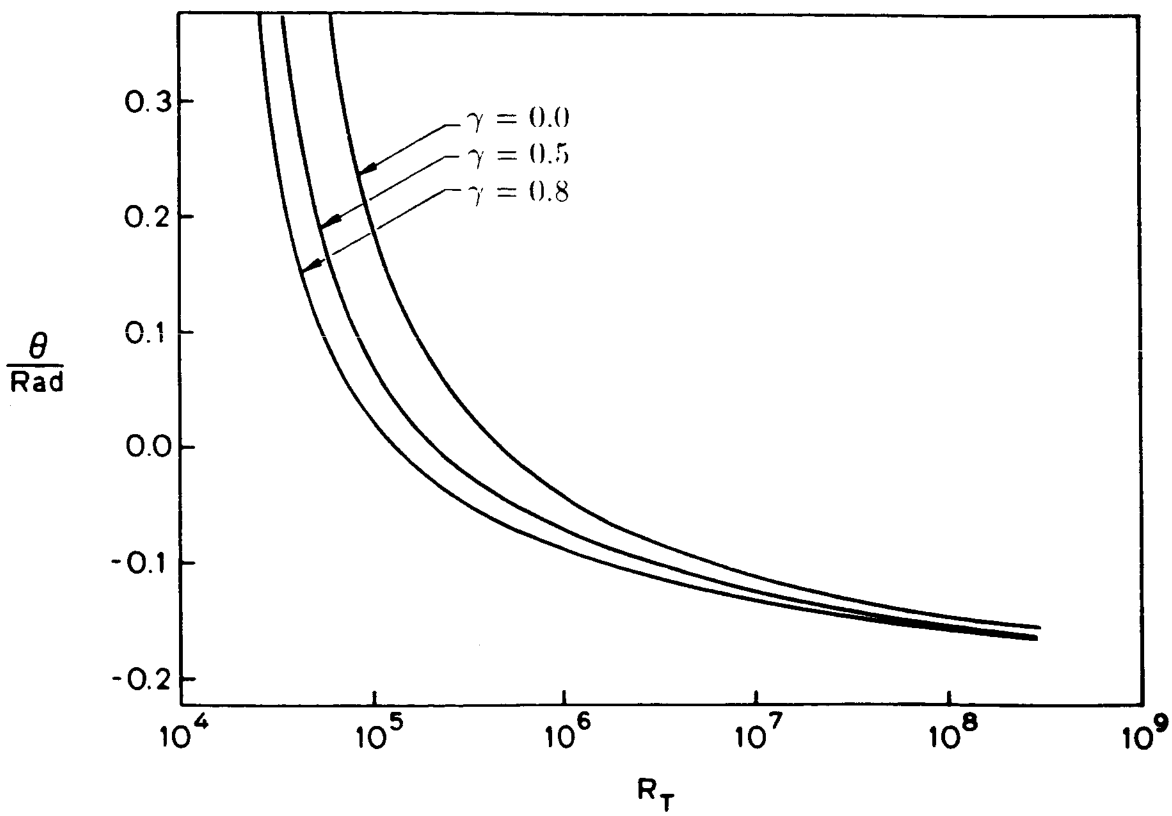


Figure 7.21: Optimal lip rotation angle as a function of Reynolds number

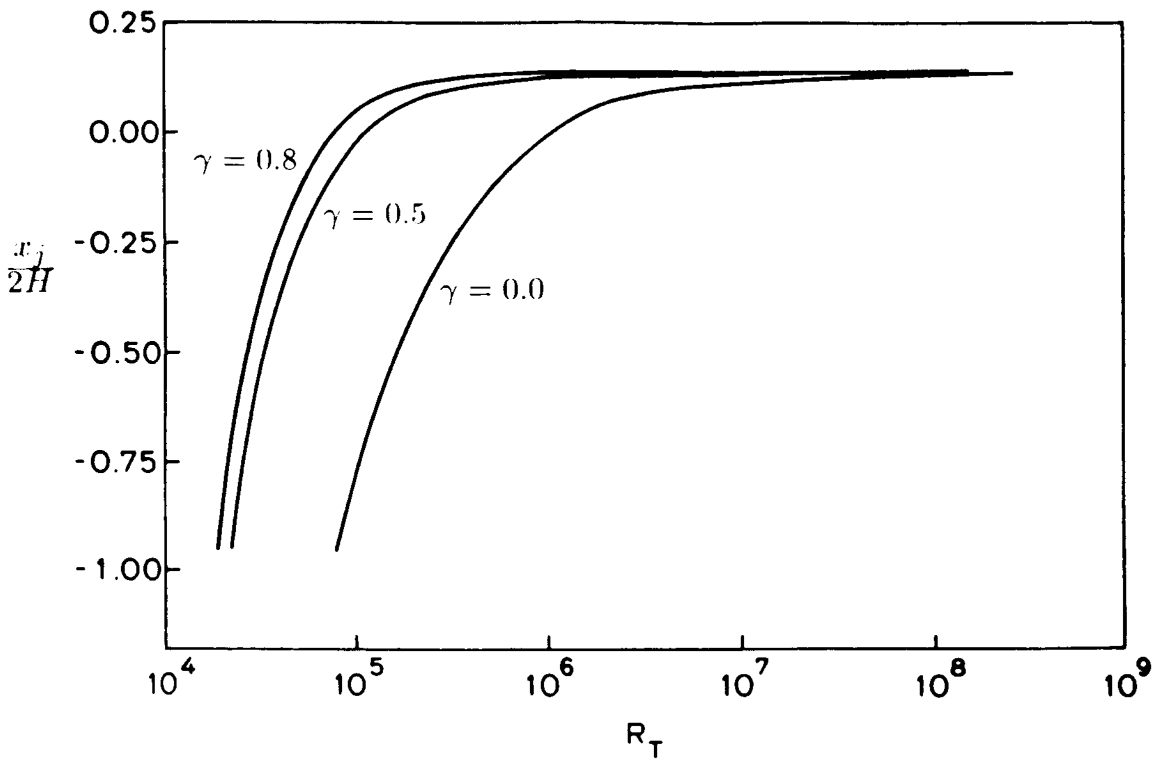


Figure 7.22: Optimal primary nozzle position as a function of Reynolds number

from large positive angles to a position of roughly zero angle. Further increase in the Reynolds number causes a continual gradual decline in the lip rotation angle. Differences in the optimal lip rotation angle due the free stream speed become increasingly small in the high Reynolds number regime.

Displayed in Figure 7.22 is the optimal primary nozzle location as a function of Reynolds number for the three values of the dimensionless free stream speed. The trends are qualitatively similar for each of the three values. In the low Reynolds number limit, the nozzle is located well in front of the shroud due to the fragile nature of the laminar boundary layers. As the Reynolds number is increased and the boundary layers become turbulent, the optimal nozzle position moves quickly to a limiting point just inside the shroud. In light of the forward stagnation point

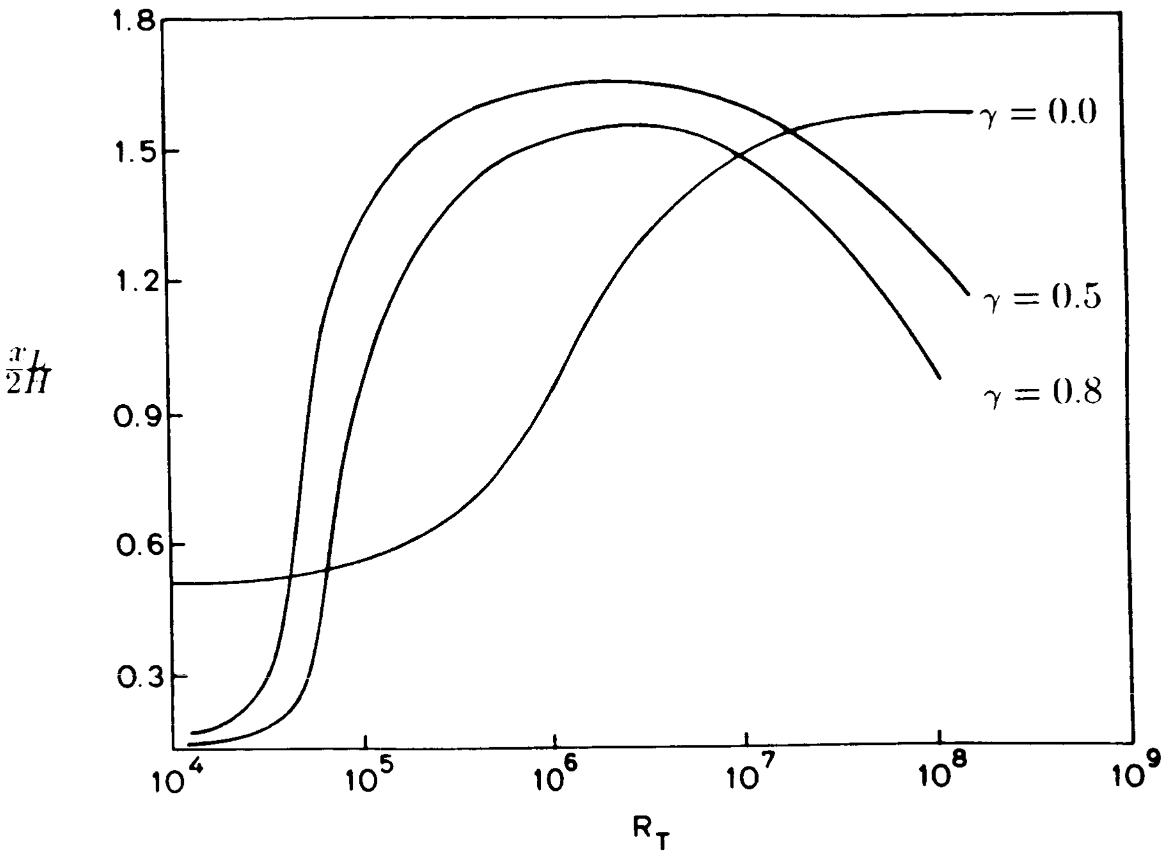


Figure 7.23: Optimal inlet lip length as a function of Reynolds number

induced by the free stream and its positive effect on the boundary layer development, the optimal nozzle location moves forward more quickly when a free stream is present as compared to static operation.

Figure 7.23 illustrates the optimal length of the inlet lip plotted as a function of Reynolds number for different values of the dimensionless free stream velocity. The general trend of a short lip at low Reynolds number, maximum lip length at moderate Reynolds number and a decline in lip length with very large Reynolds number is seen to hold for all three values of the dimensionless free stream velocity. Again due to the presence of a forward stagnation point, there is a shift in Reynolds number when the results for static operation are compared with those for a non-zero free stream. The rapid change in the lip length when moving out of the low Reynolds number regime is due to boundary layer transition.

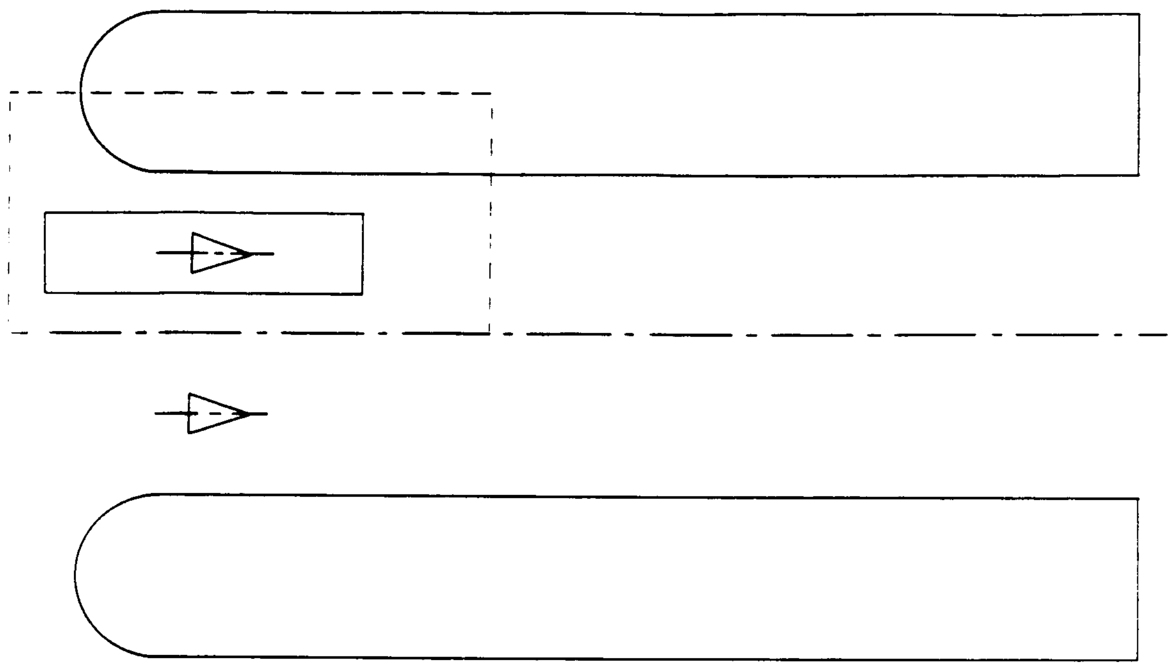


Figure 7.24: Dual-jet ejector optimization.

7.3.2 Dual-jet Ejector Optimization

The dual-jet ejector code has been used to optimize the nozzle location and tilt for the same configuration used in the parametric studies. Figure 7.24 shows the basic configuration. The optimization is performed in the following way. With the nozzle tilt fixed, the performance is computed for several different nozzle positions within the solid rectangular box shown in Figure 7.24. The resulting data is used to construct contour plots that show lines of constant thrust augmentation. The optimal nozzle position is then found simply through inspection of the contour maps. The results of the optimization study are shown in Figures 7.25-7.28. Three contour plots are shown in each of the figures, corresponding to nozzle tilts of -5^0, 0^0, and 5^0. In order to give a sense of scale, the portion of the ejector shroud contained within the dashed box in Figure 7.24 is included with the results.

Figure 7.25 shows the results for the basic ejector configuration. The most

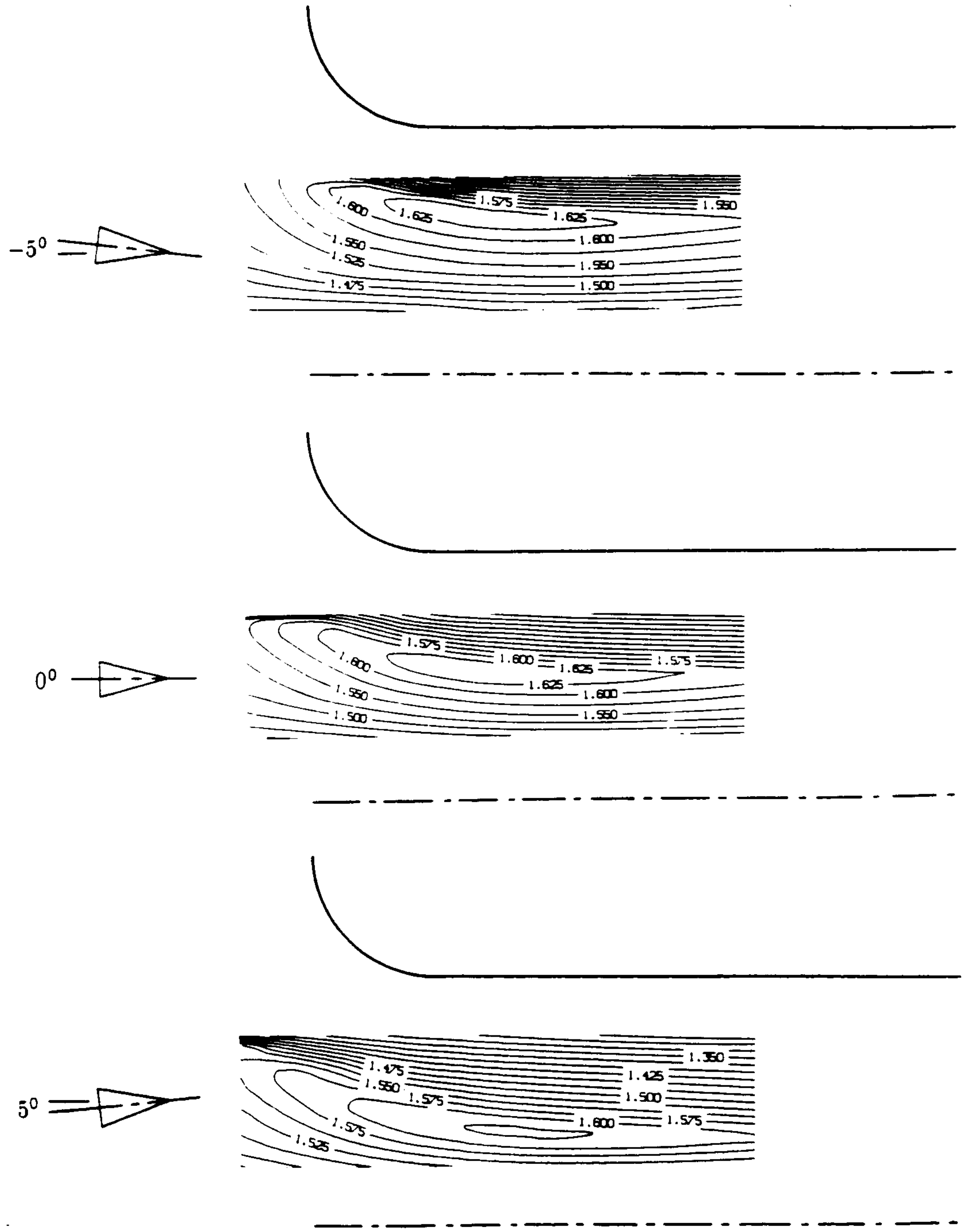


Figure 7.25: Lines of constant thrust augmentation for the unperturbed ejector. $L/2H = 3.25$, $\gamma = 0$, $\beta = 0$.

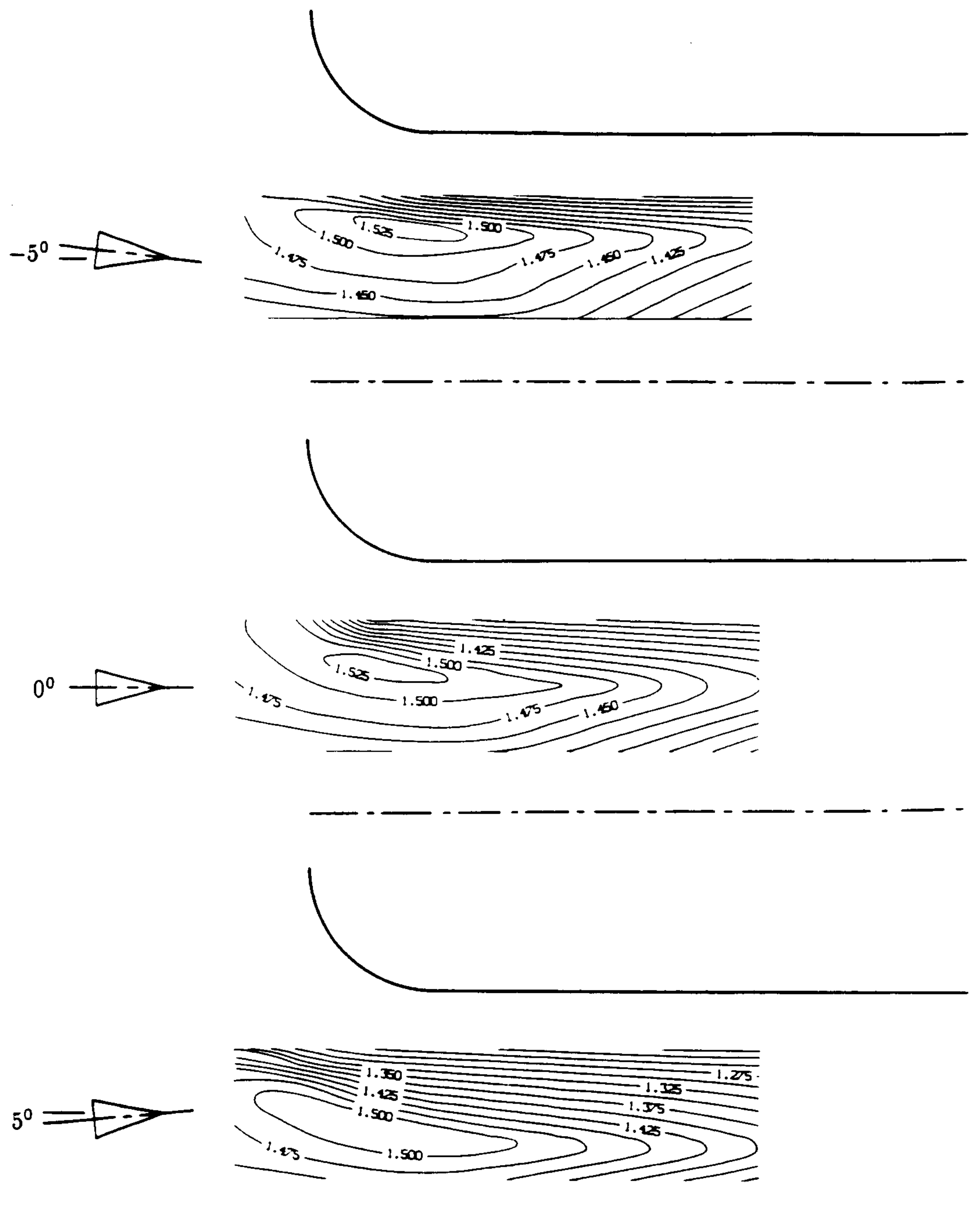


Figure 7.26: Lines of constant thrust augmentation for a shortened ejector. $L/2H = 2.25$, $\gamma = 0$, $\beta = 0$.

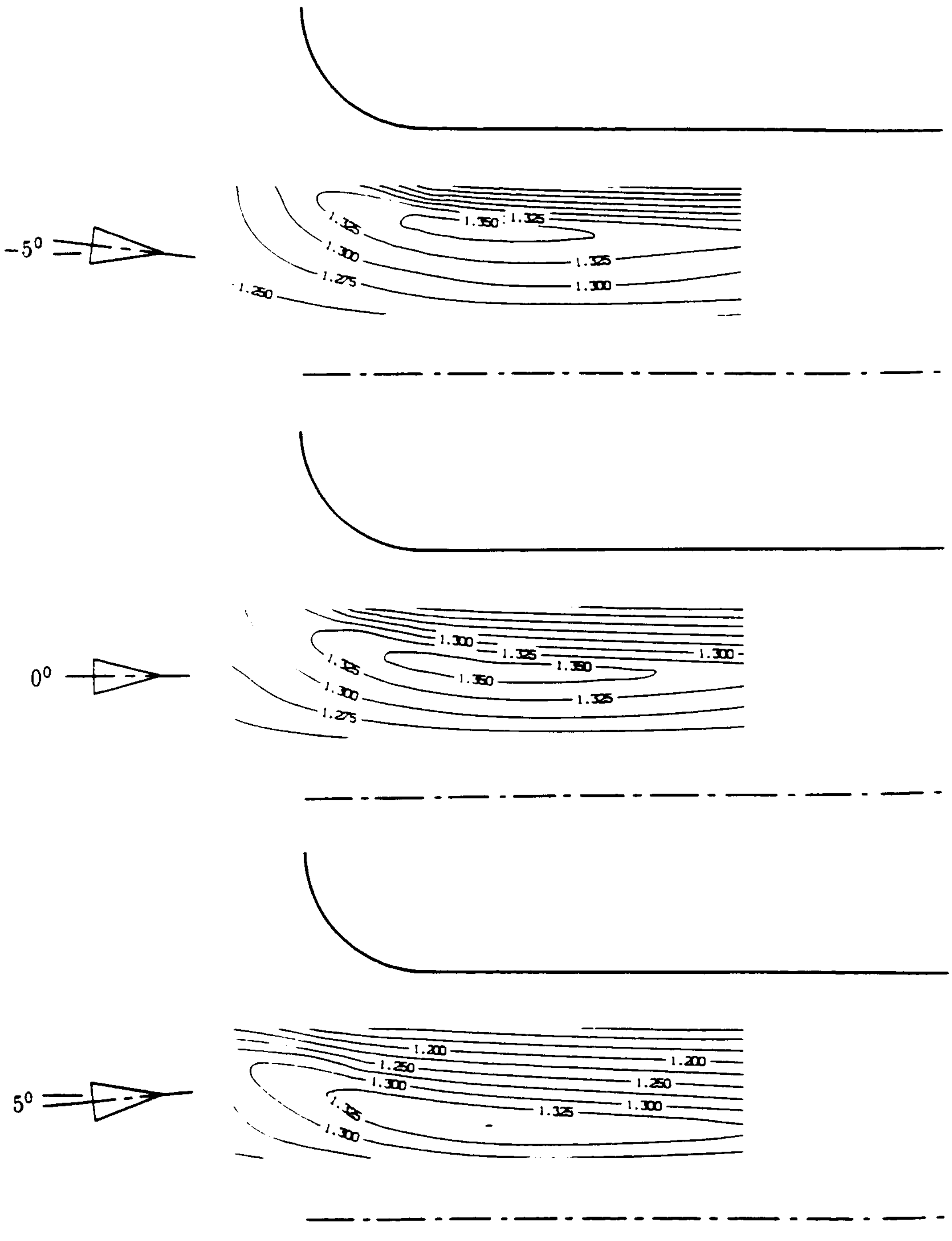


Figure 7.27: Lines of constant thrust augmentation for a moderate free stream speed. $L/2H = 3.25$, $\gamma = 0.5$, $\beta = 0$.

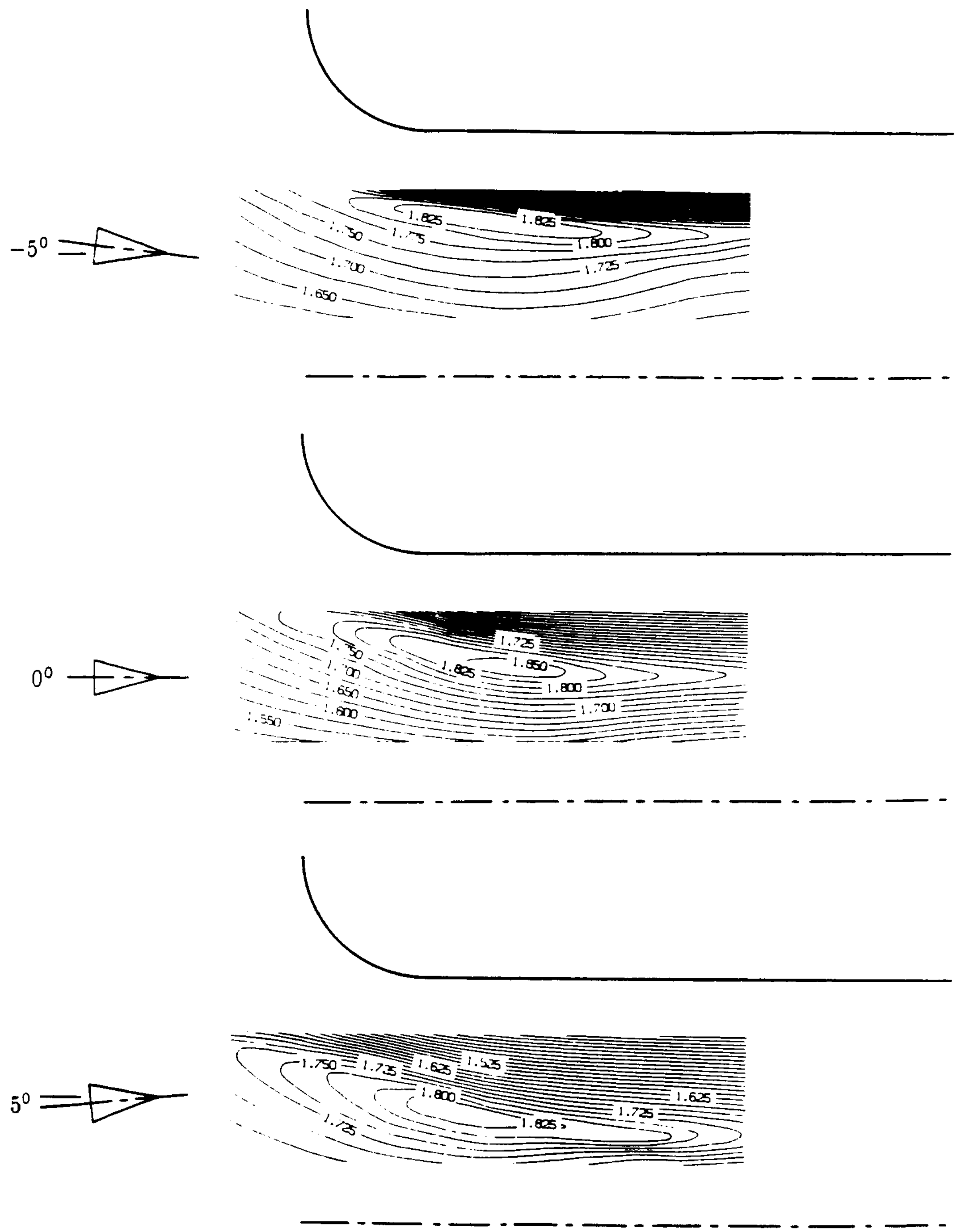


Figure 7.28: Lines of constant thrust augmentation for a moderate diffuser area ratio. $L/2H = 3.25$, $\gamma = 0$, $L_D/L = 0.7$, $\beta = 20$

obvious feature of these results is that the performance is much more sensitive to the lateral position of the nozzles than it is to the longitudinal position. The results also indicate that the optimal lateral position of the nozzles is a function of both the longitudinal position and the nozzle tilt. For each of the three tilt angles, the optimal nozzle position has a different location. As the nozzles are rotated towards each other, the optimal nozzle location moves out towards the ejector inlet and up towards the channel wall. There is little variation in the maximum thrust augmentation ratio achieved in these three cases.

Figure 7.26 shows a similar set of results for a shorter ejector ($L/2H = 2.25$). The largest difference between these results and those for a longer ejector is that the performance has become more equally sensitive to the lateral and longitudinal nozzle positions. This is primarily due to the fact that the length over which the flow has to mix has a stronger impact on performance when the latter is small (c.f. Figure 7.5). The absolute values of the thrust augmentation have also dropped in response to shortening the ejector.

Displayed in Figure 7.27 are performance contours for the basic ejector when a free stream is present. With the exception of an overall drop in performance, the results differ little from the static case shown in Figure 7.25.

Figure 7.28 shows lines of constant thrust augmentation for an ejector with a diffuser. The results show that the presence of the diffuser enhances the sensitivity of the nozzle location. This is primarily due to the fact that the effectiveness of the diffuser is a strong function of the degree of mixing achieved prior to the diffuser (c.f. Figure 2.3). Figure 7.29 shows a qualitative comparison of the computed results with experimental data[67] for the effect of nozzle position on the performance of a dual jet ejector. The experiment shows the same trend of the lateral position of the nozzle having a greater impact on performance than does the longitudinal position. The relative position of the optimal location is also similar. The absolute values of the thrust augmentation found in the experiment are higher than the computed values because a high area ratio diffuser was attached to the experimental configuration.

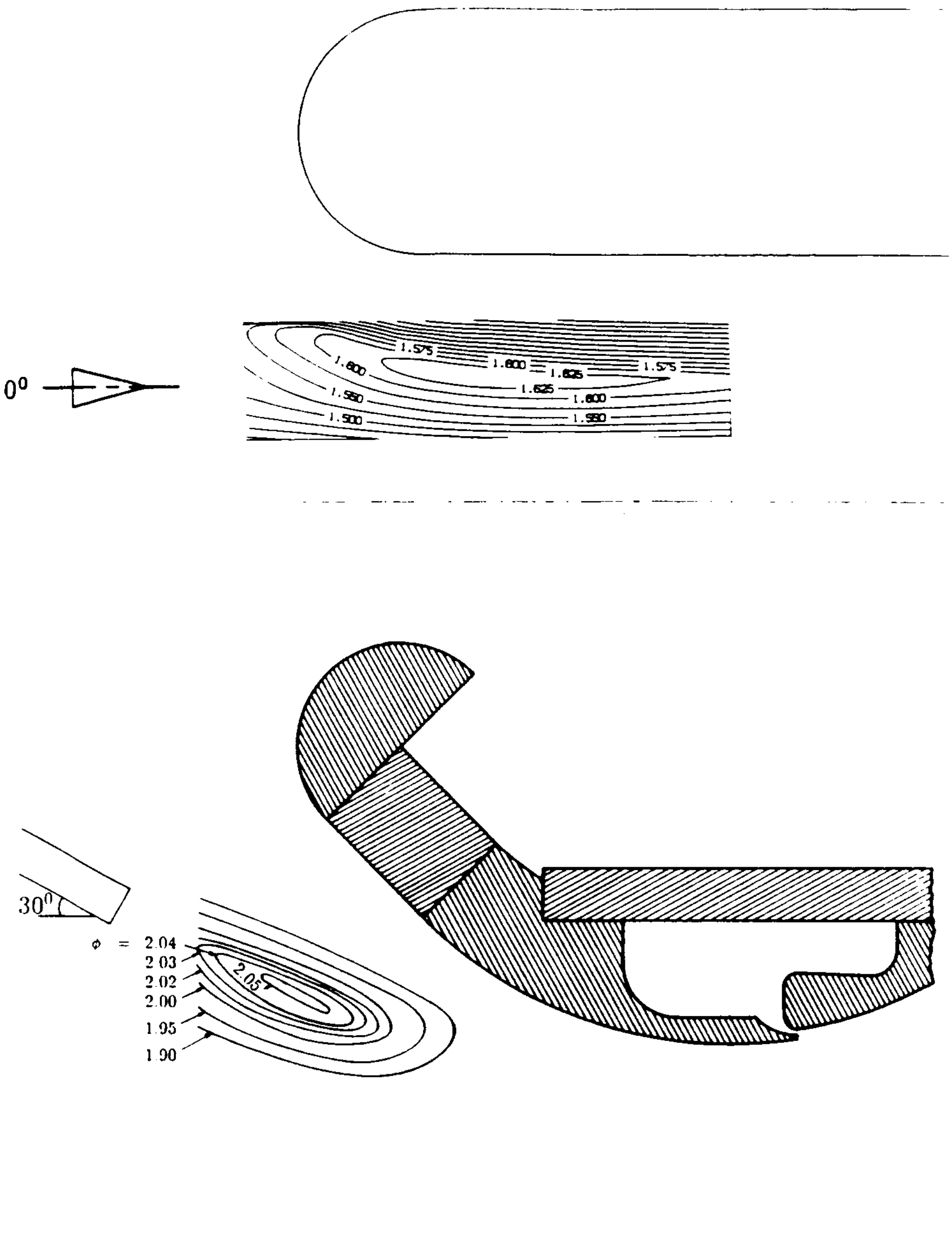


Figure 7.29: Qualitative comparison with experiment for the Dual-jet ejector nozzle position. Computation: $L/2H = 3.25$, $\gamma = 0$, $\alpha = 0$, $\beta = 0$. Experiment: $L/2H = 2.25$, $\gamma = 0$, $\alpha = -30^0$, $\beta = 45^0$.

Chapter 8

Conclusions and Recommendations

8.1 Summary

A viscous-inviscid methodology has been developed as an accurate and efficient means of evaluating the performance of thrust augmenting ejectors. The inviscid portion of the flow field is modeled with a higher order panel method, while an integral method is used to solve for the viscous jet flow. The two solutions are iteratively matched together in a process that allows each region to influence the other en route to a converged solution.

Two separate algorithms are developed; one is capable of treating ejectors with a single primary jet while the other is designed to treat configurations that use two primary jets. The results of the single-jet model compare well with experimental data. Lack of detailed experimental data for a dual-jet configuration prohibits a critical comparison to be made for this case.

Both the single and dual-jet algorithms are used in a parametric study where the influence of nozzle placement, ejector length, free stream speed, and a diffuser are investigated. The results of this study are in good qualitative agreement with the available experimental data.

The efficiency of the algorithms are demonstrated through two optimization problems. For the single-jet ejector, the nozzle position and the inlet shape are optimized for various flight speeds and Reynolds numbers. The dual-jet ejector algorithm is used to optimize the lateral and longitudinal nozzle position for different nozzle tilt angles.

8.2 Conclusions of the Numerical Method

Viscous-inviscid algorithms have been successfully developed to model single-jet and dual-jet ejector flow fields. The main conclusions that have been arrived at in connection with the use of this numerical technique are as follows:

1. The viscous-inviscid technique yields accurate solutions. Predictions of the model agree well with experimental data.

2. The viscous-inviscid technique is efficient. The computing time required for a solution is roughly 1.5 and 3 minutes of CPU time for the single-jet and dual-jet algorithms respectively on a VAX 11/780 machine.

3. The viscous-inviscid technique is robust in its ability to model arbitrary symmetric ejector configurations. This fact is demonstrated in the parametric studies.

4. The viscous-inviscid technique is well suited as for thrust augmentor optimization work.

8.3 Conclusions of the Parametric Studies

The parametric studies predict how the thrust augmentor performance is affected by the details of the ejector shape. The main conclusions of the parametric studies are as follows:

1. In all cases the dual-jet ejector performs better than the single-jet counterpart. The dual-jet ejector improvement is substantial; thrust augmentation ratio

increases of 20% to 50% can be realized by replacing a single primary jet with two primary nozzles.

2. The performance is maximized when the primary nozzle is located at the entrance plane of the ejector for the single-jet configuration. For the dual-jet ejector the performance is maximized when the jet nozzles are placed slightly inside of the ejector.

3. For the dual-jet ejector, the performance is maximized when the jet trajectories are such that the jet centerlines remain equi-spaced between the ejector symmetry plane and the ejector wall.

4. The thrust augmentation ratio increases with increasing ejector length. For short ejectors, the performance of the dual-jet ejector increases more rapidly with length than does the single-jet configuration.

5. Thrust augmentor performance degrades rapidly with increasing free stream speed.

6. The inclusion of a diffuser improves the ejector performance. The dual-jet ejector benefits more greatly from a diffuser than does the single-jet configuration.

7. In the absence of separation, the details of the shape of the diffuser are relatively unimportant. The thrust augmentation ratio is primarily a function of the diffuser area ratio alone.

8.4 Conclusions of the Optimization Studies

Optimization studies were performed to demonstrate the efficiency of the viscous-inviscid algorithms. The main conclusions of these studies are:

1. Boundary layer separation is a controlling factor in the design of an ejector inlet.

2. An ejector needs a variable-geometry inlet to maintain optimal performance in all flight regimes.

3. Both the optimal longitudinal and lateral position of the primary nozzles in a dual-jet ejector are a function of the nozzle tilt angle.

4. As the nozzles are tilted towards each other, the optimal nozzle position moves towards the ejector wall and out towards the ejector inlet.

8.5 Recommendations

The work presented here should be considered as the first step in creating a general, efficient procedure for modeling the ejector mixing problem. There are several extensions of this work that are necessary to achieve the ultimate goal. These are:

1. Extend the analysis to account for the effects of compressibility. To do this, both a temperature profile and a thermal energy equation will need to be included in the integral formulation for the viscous region. For the inviscid region, a compressibility correction to the panel method (such as the Prandtl-Glauert correction) could be used if the secondary flow is purely subsonic. If a supersonic secondary flow is to be modeled, a finite difference solution to either the full potential equation or the Euler equations will be necessary.

2. Remove the point source of momentum approximation for the primary jet and replace it with a more realistic finite-width model. This step will allow the effect of the nozzle width to be determined and should make the overall results more accurate by taking into account the jet potential core region.

3. Investigate the use of more sophisticated turbulence models. The algebraic eddy-viscosity expression used here appears to be adequate, but is limited in its rough approximation of the turbulent transport process. Other approaches, such as the $k - \epsilon$ model, are based on a more realistic picture of turbulence. Use of a model of this type should improve the reliability of the results.

4. Extend the analysis to three or more primary jets. When this step is undertaken, it should be done in conjunction with a finite-width jet nozzle model. It is necessary to use a finite nozzle model to properly account for the secondary flow blockage that results from placing additional nozzles within the ejector inlet.

5. Ultimately, the model should be extended to three-dimensional flows. A three-dimensional analysis would be a valuable aid in the design of compact ejectors of low aspect ratio.

Appendix A

Compressibility of the Secondary Flow

In this appendix, some of the limitations of the incompressible flow assumption are investigated. This investigation is necessary since most ejectors are designed to operate in the compressible flow regime. The analysis contained here illustrates that the thickness of the ejector shroud and the jet exit Mach number are important parameters in ascertaining the extent to which the secondary flow is incompressible.

The analysis is begun with the definition of the thrust augmentation ratio:

$$\begin{aligned}\phi &= \frac{T_0 + T_i}{T_0} \\ &= 1 + \frac{T_i}{T_0} \qquad \text{(A.1)}\end{aligned}$$

where T_0 is the primary nozzle thrust and T_i is the thrust induced by the suction acting over the leading edges of the shroud (see Figure A.1). The induced thrust may be written as

$$T_i = 2\int_0^d (p_{atm} - p)dy \qquad \text{(A.2)}$$

where the factor of 2 accounts for both leading edges of the ejector shroud. Let the average pressure acting over the leading edges be denoted as $\bar{p}_{le}$. Then

$$\bar{p}_{le} = \frac{1}{d}\int_0^d pdy \qquad \text{(A.3)}$$

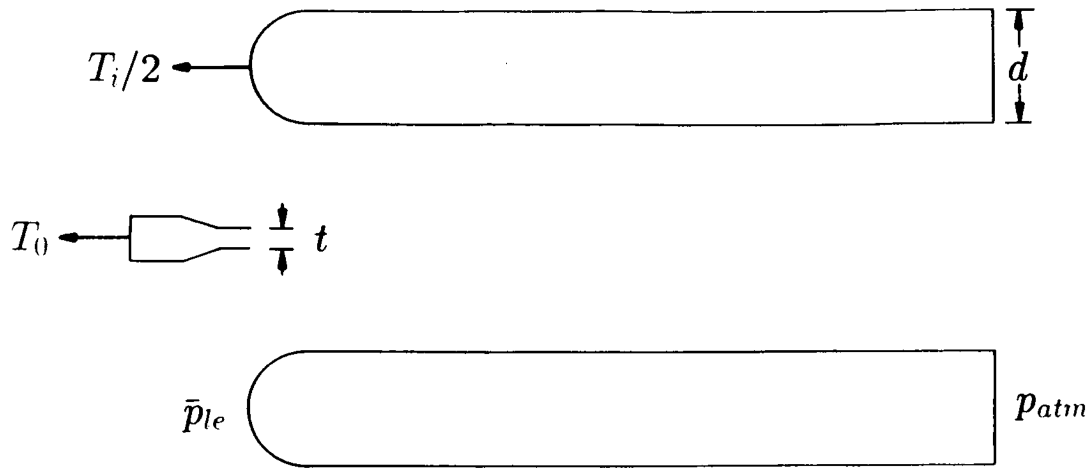


Figure A.1: Ejector geometry and the principle of thrust augmentation.

The induced thrust may be written in terms of the average leading edge suction by combining Eqs. (A.2) and (A.3)

$$T_i = 2(p_{atm} - \bar{p}_{le})d \tag{A.4}$$

This result is combined with the expression for the thrust augmentation ratio given in Eq. (A.1) to give

$$\begin{aligned} \phi &= 1 + 2\frac{(p_{atm} - \bar{p}_{le})d}{\rho_{ex} u_{ex}^2 t} \\ &= 1 + 2\frac{p_{atm}}{\rho_{ex} u_{ex}^2}\left(1 - \frac{\bar{p}_{le}}{p_{atm}}\right)\left(\frac{d}{t}\right) \end{aligned} \tag{A.5}$$

where the primary jet thrust has been rewritten in terms of the exiting momentum flux. Assume that the jet nozzle is designed to fully expand the primary flow to the atmospheric pressure. In this case, the definition of the sound speed, $c^2 = \gamma p/\rho$, can be used in Eq. (A.5) to give

$$\phi = 1 + \frac{2}{\gamma}\left(\frac{c_{ex}}{u_{ex}}\right)^2 \left(1 - \frac{\bar{p}_{le}}{p_{atm}}\right)\left(\frac{d}{t}\right) \tag{A.6}$$

or in terms of the Mach number

$$\phi = 1 + \frac{2}{\gamma M_{ex}^2}\left(1 - \frac{\bar{p}_{le}}{p_{atm}}\right)\left(\frac{d}{t}\right) \tag{A.7}$$

A.1 Magnitude of the Leading Edge Suction

Equation (A.7) can be used to determine the magnitude of the average leading edge suction for given values of the thrust augmentation, exit Mach number, and non-dimensional shroud thickness. To investigate the magnitude of the leading edge suction further, Eq. (A.7) is rewritten as

$$\frac{\bar{p}_{le}}{p_{atm}} = 1 - \left(\phi - 1\right)\frac{\gamma M_{ex}^2}{2}\left(\frac{t}{d}\right) \tag{A.8}$$

For the purpose of illustration assume that $\gamma = 7/5$ and $\phi = 2.0$. The above relation then becomes

$$\frac{\bar{p}_{le}}{p_{atm}} = 1 - \frac{7}{10}M_{ex}^2\left(\frac{t}{d}\right) \tag{A.9}$$

Figure A.2 shows the magnitude of the leading edge suction predicted by the above equation as a function of the jet exit Mach number, with the non-dimensional shroud thickness appearing as a parameter. The plot shows that for an extremely thin shroud ($d/t = 1$), the leading edge pressure drops rapidly with increasing exit Mach number. For this value of shroud thickness, the average leading edge pressure is one half the atmospheric value at $M_{ex} = 0.85$, and is required to be vacuum at $M_{ex} = 1.2$. As the shroud thickness is increased, the leading edge suction decreases so that the force developed on the ejector shroud is constant (i.e. constant thrust augmentation has been assumed). For moderate shroud thickness ($d/t = 5.0$), the leading edge pressure drops below one half atmosphere at $M_{ex} = 1.9$ and is required to be vacuum at $M_{ex} = 2.7$. For a thicker shroud ($d/t = 10.0$), the leading edge suction is moderate for low Mach numbers. The leading edge pressure falls to one half atmosphere at $M_{ex} = 2.7$ and vacuum at $M_{ex} = 3.8$.

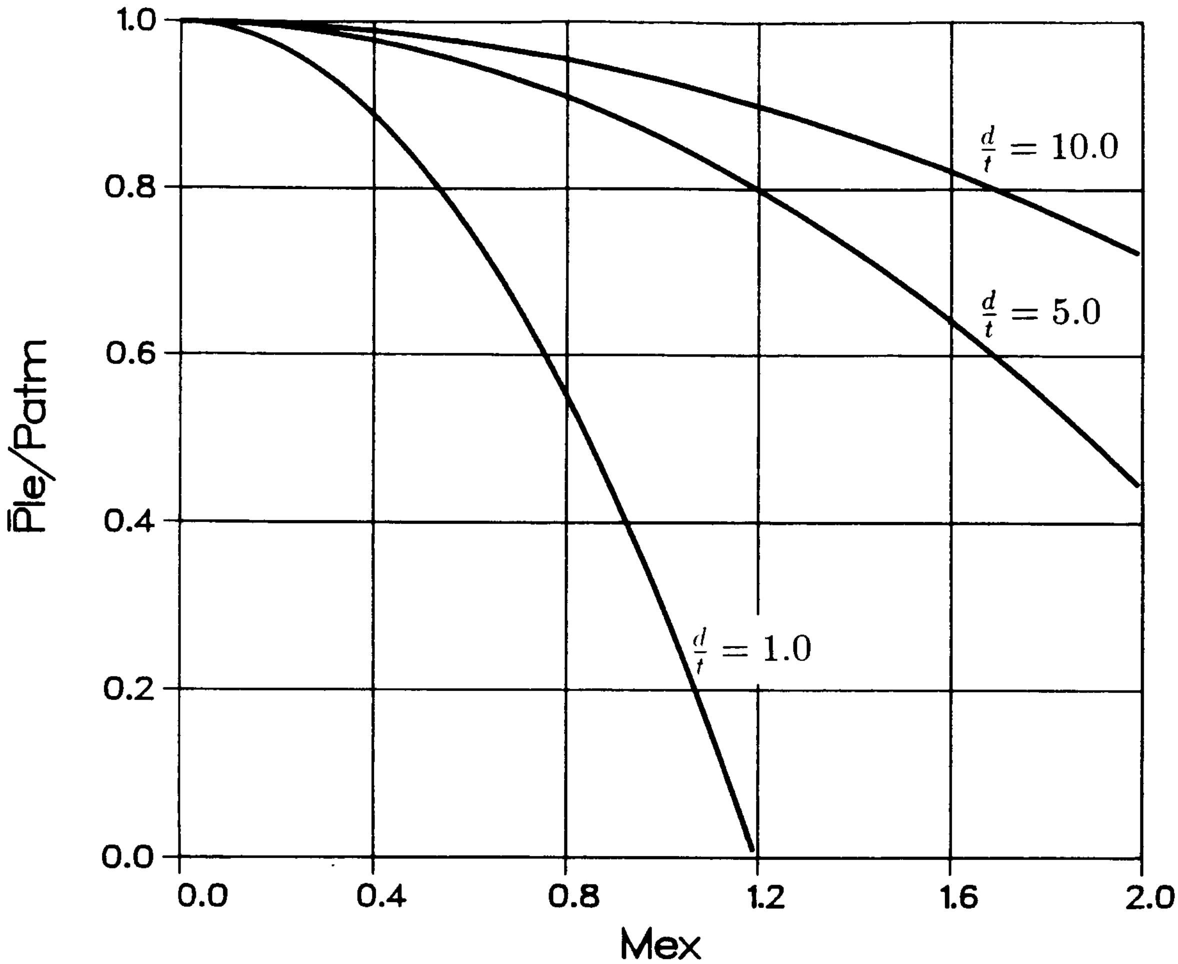


Figure A.2: Magnitude of the shroud leading edge suction as a function of the jet exit Mach number. $\gamma = 7/5$ and $\phi = 2.0$.

A.2 Conditions for Effectively Incompressible Flow

Another useful feature of this analysis is that it can be used to give the conditions under which an assumption of incompressible secondary flow is valid. This is done by solving Eq. (A.7) for d/t:

$$\frac{d}{t} = \frac{(\phi - 1)}{\left(1 - \frac{\bar{p}_{le}}{p_{atm}}\right)} \frac{\gamma M_{ex}^2}{2} \tag{A.10}$$

For the purpose of illustration, assume that compressible effects become important in the secondary flow when the Mach number at the shroud leading edge is greater than 0.3. The isentropic relation

$$\frac{p_{atm}}{\bar{p}_{le}} = \left[1 + \left(\frac{\gamma - 1}{2}\right) M_{le}^2\right]^{\frac{\gamma}{\gamma-1}} \tag{A.11}$$

indicates that $M_{le} = 0.3$ corresponds to a leading edge pressure of $\frac{\bar{p}_{le}}{p_{atm}} = 0.9395$ (for $\gamma = 7/5$). With this value of the leading edge pressure, together with $\gamma = 7/5$ and $\phi = 2.0$, Eq. (A.10) becomes

$$\frac{d}{t} = 11.56 M_{ex}^2 \tag{A.12}$$

For a given jet exit Mach number, this equation gives the minimum shroud thickness required to ensure that the leading edge Mach number is less than 0.3 for $\gamma = 7/5$ and $\phi = 2.0$. Figure A.3 shows a plot of the boundary predicted by Eq. (A.12). The results show that the shroud thickness must increase with increasing jet exit Mach number in order to keep the leading edge Mach number within the effectively incompressible range. If the results of an ejector analysis that assumes incompressible secondary flow are to be used, then the combination of jet exit Mach number and non-dimensional shroud thickness must lie above the bounding curve in Figure A.3.

A.3 Conclusions

The results of this study indicate that the thickness of the ejector shroud is an important parameter in ejector design. In order to achieve a desired level of thrust

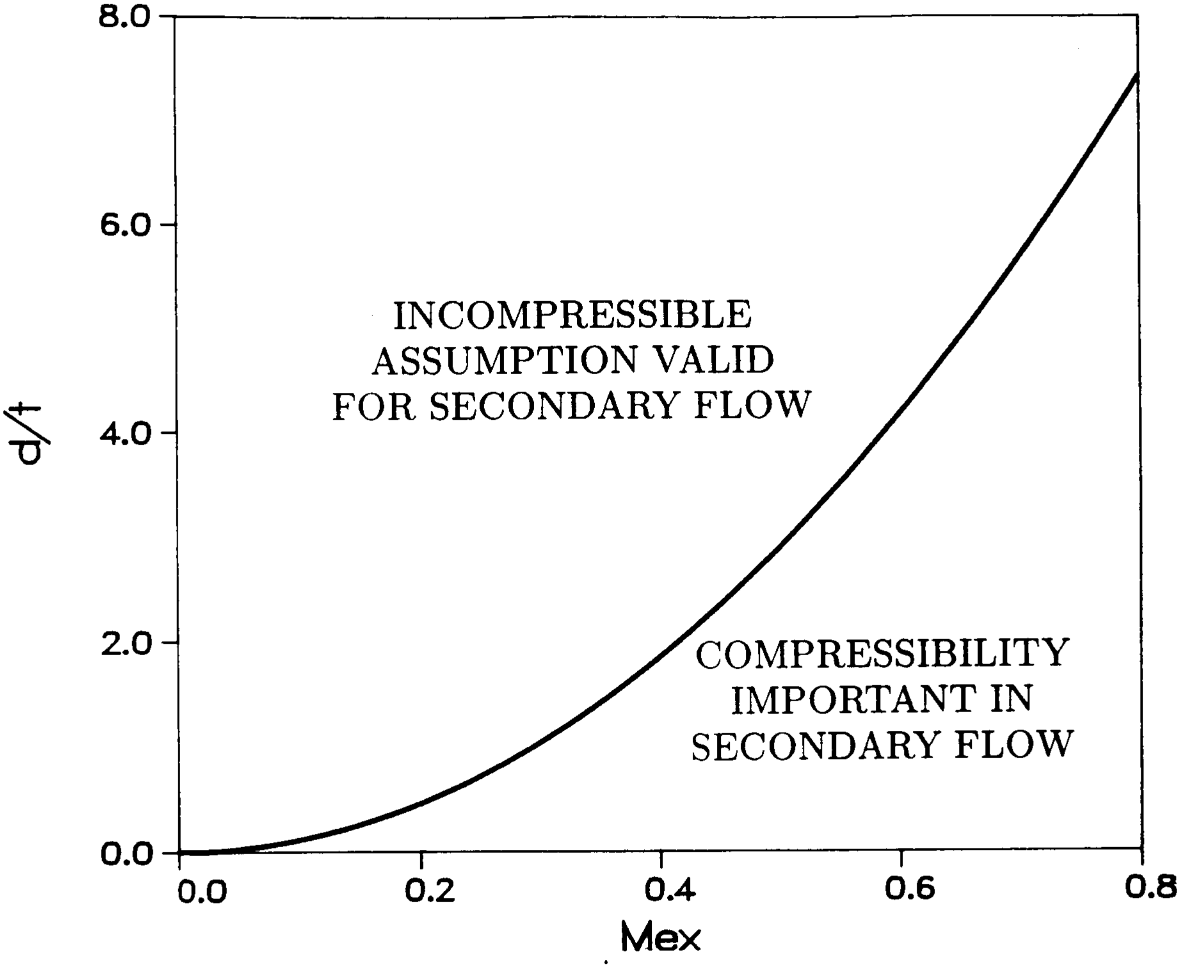


Figure A.3: Boundary for the incompressible flow assumption. $\gamma = 7/5$, $\phi = 2.0$.

augmentation at a given primary jet exit Mach number, the ejector shroud must be sufficiently thick so that the leading edge pressure is not required to be non-physically small. In addition, if the results of an incompressible analysis are used in ejector design, the shroud must be sufficiently thick so that the secondary flow remains effectively incompressible for the given operating jet exit Mach number.

Appendix B

Computer Code

This appendix contains source listings for both the single-jet and dual-jet viscous-inviscid algorithms. The various subroutines are grouped into four libraries: AUGLIB, TWINLIB, PAN2LIB, and MATHLIB. The AUGLIB library contains the subroutines for the single-jet viscous-inviscid matching procedure. The TWINLIB contains the subroutines for the dual-jet viscous-inviscid matching procedure. The PAN2LIB contains the subroutines needed to compute the higher-order panel method. Finally, the MATHLIB contains various mathematics procedures. In addition to these libraries, the IMSL library is used to supply several mathematics routines.

Both the single-jet and dual-jet codes have undergone revisions since the time that the results shown in this report were generated. Because of this, the code shown in this appendix may produce results that differ slightly from those contained within the results section.

B.1 Single-Jet Program AUGMENT

AUGMENT is the driving program for the single-jet viscous-inviscid algorithm. Once compiled, it must be linked with the AUGLIB, PAN2LIB, MATHLIB, and IMSL libraries. Input data are to be read from file CASE.DAT.


```
      PROGRAM AUGMENT
C
C*****************************************************************************
C                                                                            *
C     PROGRAM AUGMENT COMPUTES THE PERFORMANCE OF A TWO-DIMENSIONAL SINGLE-JET*
C INCOMPRESSIBLE FLOW EJECTOR.  THE CODE IS BASED ON A VISCOUS-INVISCID INTER-*
C ACTION ALGORITHM IN WHICH THE INVISCID REGION IS COMPUTED WITH A HIGHER    *
C ORDER PANEL METHOD AND THE VISCOUS ZONE IS COMPUTED WITH AN INTEGRAL METHOD.*
C     INPUT DATA IS READ FROM FILE CASE.DAT.  THE ITERATION HISTORY AS WELL  *
C AS THE THRUST AUGMENTATION RATIO PREDICTION ARE WRITTEN TO FILE OUT.DAT.   *
C AN EXTENDED OUTPUT OPTION MAY BE SPECIFIED IN THE INPUT DATA FILE TO CAUSE *
C THE JET SOLUTION AS WELL AS THE DETAILS OF THE MATCHING HISTORY TO BE      *
C OUTPUT.                                                                    *
C     THIS PROGRAM MUST BE LINKED TO THE MATHLIB AND PAN2LIB LIBRARIES AS WELL*
C AS THE IMSL MATH LIBRARY.                                                  *
C     THIS CODE IS OF EVOLUTIONARY ORIGIN AND CONSEQUENTLY MAY CONTAIN REGIONS*
C POOR LOGIC STRUCTURE AND INEFFICIENT PROCEDURES.  THERE HAS BEEN NO ATTEMPT *
C MADE TO UPGRADE THE CODE TO A "PRODUCTION CODE" STATUS.                    *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(MAX=300)
      DIMENSION XBOD(MAX),YBOD(MAX),VN(MAX),XCP(MAX),YCP(MAX),
     &          ALPHA(MAX),D(MAX),ZETA(MAX),CX(3*MAX),CY(3*MAX),
     &          PD(MAX),PE(MAX),PF(MAX),PG(MAX),PH(MAX),PPI(MAX),
     &          C(MAX),IND1(MAX),IND2(MAX),A(MAX),B(MAX),AMAT(MAX,MAX),
     &          BMAT(MAX,MAX),WORK(8*MAX),W(MAX*MAX),WINV(MAX*MAX),
     &          Q(MAX),AJET(100*MAX),BJET(100*MAX),XJET(50),YJET(50),
     &          UJET(50),VJET(50),R(5),
     &          XS(250),VS(250),SC(100),UEXT(100)
      LOGICAL DUMP1,STAG,DUMP,SEP,BLAYER
C
C        *** OPEN DATA FILES.  BODY.DAT WILL CONTAIN THE COORDINATES OF THE  ***
C        *** EJECTOR SHROUD.  PARAM.DAT CONTAINS THE FREE STREAM VELOCITY AS ***
C        *** WELL AS THE ANGLE OF ATTACK.  CASE.DAT CONTAINS THE INPUT DATA. ***
C        *** OUT.DAT CONTAINS THE CONVERGENCE HISTORY AS WELL AS THE THRUST  ***
C        *** AUGMENTATION RATIO.                                             ***
C
      OPEN(UNIT=1,NAME='BODY.DAT',TYPE='NEW',FORM='FORMATTED')
      OPEN(UNIT=2,NAME='PARAM.DAT',TYPE='NEW',FORM='FORMATTED')
      OPEN(UNIT=4,NAME='CASE.DAT',TYPE='OLD',FORM='FORMATTED')
      OPEN(UNIT=21,NAME='OUT.DAT',TYPE='NEW',FORM='FORMATTED')
C
C         *** TOL1 IS THE CONVERGENCE TOLERANCE FOR THE VISCOUS-INVISCID    ***
C         *** MATCHING, TOL2 IS THE CONVERGENCE TOLERANCE FOR THE EXIT      ***
C         *** PRESSURE MATCHING.  BO IS THE JET INITIAL HALF-WIDTH.         ***
C
      TOL1=5.0D-3
      TOL2=5.0D-3
      BO=1.D-2
C
C         *** WRITE THE FREE STREAM VELOCITY AND ANGLE OF ATTACK TO FILE    ***
C         *** PARAM.DAT.  THE FREE STREAM VELOCITY WILL BE REDEFINED LATER  ***
C         *** IF IT IS TO BE NON-ZERO.                                      ***
C
      V0=0.0D0
      BETA=0.0D0
      WRITE(2,5) V0,BETA
5     FORMAT(2F10.5)
      REWIND 2
```

```
C
C       *** READ INPUT PARAMETERS FROM FILE CASE.DAT.                    ***
C
      CALL GETPRM(XJ,XLIP,THLIP,XEXIT,XDIF,DIFSLP,GAMMA,U10,DUMP1,
     &            BLAYER,RE)
C
C       *** IF THE EXTENDED OUTPUT OPTION IS CHOSEN, OPEN ADDITIONAL FILES.***
C       *** VLCJET.DAT CONTAINS THE VELOCITY COMPONENTS ALONG THE JET      ***
C       *** BOUNDARY.   MCHJET.DAT CONTAINS THE JET SOLUTION WITHIN THE    ***
C       *** VISCOUS-INVISCID MATCHING REGION.   CHNJET.DAT CONTAINS THE    ***
C       *** JET SOLUTION WITHIN THE FULLY VISCOUS REGION.                  ***
C
      IF(DUMP1) THEN
         OPEN(UNIT=9,NAME='VELJET.DAT',TYPE='NEW',FORM='FORMATTED')
         OPEN(UNIT=10,NAME='MCHJET.DAT',TYPE='NEW',FORM='FORMATTED')
         OPEN(UNIT=12,NAME='CHNJET.DAT',TYPE='NEW',FORM='FORMATTED')
      END IF
C
C       *** GENERATE THE EJECTOR GEOMETRY AND WRITE THE COORDINATES TO THE ***
C       *** FILE BODY.DAT.                                                 ***
C
      CALL BODGEN(XJ,XLIP,THLIP,NJS,NJF,NS,NF)
      NJET=NJF-NJS+1
C
C       *** READ THE EJECTOR BODY COORDINATES AND PANEL SUCTION VELOCITIES ***
C       *** INTO ARRAYS.                                                   ***
C
      CALL GETDAT(1,XBOD,YBOD,VN,N,V0,BETA)
C
C       *** COMPUTE THE FREE STREAM VELOCITY FROM THE PARAMETER GAMMA AND  ***
C       *** THE PRIMARY JET THRUST.                                        ***
C
      CALL FRESTM(U10,B0,1.0D0,GAMMA,V0)
C
C       *** COMPUTE GEOMETRICAL PARAMETERS FOR THE PANEL METHOD            ***
C
      CALL GEOM(XBOD,YBOD,ZETA,CX,CY,WORK,N,XCP,YCP,ALPHA,D,
     &          IND1,IND2,PD,PE,PF,PG,PH,PPI,C)
C
C       *** COMPUTE THE AERODYNAMIC INFLUENCE MATRIX AND ITS INVERSE       ***
C
      CALL INFINV(XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,PF,PG,PH,PPI,C,
     &            WORK,A,B,W,N,AMAT,BMAT,WINV)
C
C      *** COMPUTE THE INFLUENCE VELOCITY COEFFICIENTS ALONG THE JET       ***
C      *** BOUNDARY.                                                       ***
C
      CALL JETCOF(NJS,NJF,XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,PF,PG,PH,PPI,
     &            C,WORK,A,B,AMAT,BMAT,N,XJET,YJET,NJET,AJET,BJET)
C
C      *** ENTER A LOOP TO DO THE EXIT PRESSURE MATCHING.                  ***
C
      JMAX=10
      DO J=1,JMAX
C
C          *** ENTER A LOOP TO DO THE VISCOUS-INVISCID MATCHING            ***
C
         IMAX=10
         DO I=1,IMAX
C
C                *** COMPUTE THE PANEL SOURCE STRENGTHS.                   ***
```

```
C
            CALL STRNTH(ALPHA,VN,WINV,N,VO,BETA,Q)
C
C              *** COMPUTE THE VELOCITY COMPONENTS ALONG THE JET BOUNDARY   ***
C
            CALL JETVEL(AJET,BJET,NJET,Q,N,VO,BETA,UJET,VJET,
     1                  UOO,PATM)
C
C              *** COMPUTE THE JET SOLUTION WITHIN THE VISCOUS-INVISCID     ***
C              *** MATCHING REGION.                                         ***
C
            CALL JET(NJS,NJF,XJET,YJET,UJET,VJET,NJET,U10,BO,
     1               VN,N,DUMP1,I,XEND,R,RES)
C
C              *** COMPUTE THE FREE STREAM VELOCITY.                        ***
C
            CALL FRESTM(U10,BO,UOO,GAMMA,VO)
C
C              *** CHECK FOR CONVERGENCE IN THE VISCOUS-INVISCID MATCHING.  ***
C
            IF(I.GT.1.AND.DABS(RES).LT.TOL1) GOTO 20
            IF(I.EQ.IMAX) THEN
              WRITE(21,10)
10            FORMAT(' VISCOUS-INVISCID MATCHING DID NOT CONVERGE')
              STOP
            END IF
C
          END DO
C
20        CONTINUE
C
C            *** COMPUTE THE JET SOLUTION WITHIN THE FULLY VISCOUS REGION   ***
C
          CALL CHANEL(R,XEXIT,XEND,XDIF,DIFSLP,DUMP1,PEXIT,DFDRAG)
C
C            *** UPDATE THE INITIAL JET VELOCITY                            ***
C
          ROLD=RR
          RR=(PATM-PEXIT)
          IF(J.EQ.1) THEN
            WW=0.2
          ELSE
            WW=-(U10-U10OLD)/(RR-ROLD)
          END IF
          U10OLD=U10
          U10=U10+WW*RR
C
C            *** WRITE CONVERGENCE INFORMATION.                             ***
C
          WRITE(21,40) PATM,PEXIT,RR,U10
40        FORMAT(' PATM = ',F10.5,' PEXIT = ',F10.5,
     &           ' R = ',F10.5,' U10 = ',F10.5)
C
C            *** CHECK FOR CONVERGENCE IN THE EXIT PRESSURE MATCHING.       ***
C
          IF(DABS(RR).LT.TOL2) GOTO 90
C
        END DO
C
90      CONTINUE
C
```

```
      IF(BLAYER) THEN
C
C           *** COMPUTE THE EJECTOR SURFACE VELOCITIES.                 ***
C
         CALL SURFVEL(XEND,XEXIT,XCP,YCP,D,AMAT,BMAT,Q,N,
     1                 V0,BETA,SC,UEXT,NEXT,XLEN,STAG)
C
C           *** COMPUTE THE BOUNDARY LAYER.                             ***
C
         DUMP=.FALSE.
         NSTEP=20
         CALL AUGLYR(SC,UEXT,NEXT,RE,STAG,DUMP,NSTEP,SEP,SCRIT)
C
C           *** WRITE THE RESULTS.                                      ***
C
         IF(SEP) THEN
            WRITE(21,100)
100         FORMAT(/,' SEPARATED BOUNDARY LAYER',/)
         ELSE
            WRITE(21,110)
110         FORMAT(/,' NO SEPARATION',/)
         END IF
      END IF
C
C        *** COMPUTE THE DIFFUSER EXIT WIDTH.                           ***
C
      HEXIT=1.0D0+(XEXIT-XDIF)*DIFSLP
C
C        *** COMPUTE THE EJECTOR PERFORMANCE                            ***
C
      CALL PERFRM(R,HEXIT,ALPHA,D,AMAT,BMAT,Q,N,V0,BETA,
     1             U10,U00,B0,DFDRAG,NS,NF,PHI)
C
C        *** REWRITE THE EJECTOR BODY GEOMETRY FILE WITH THE CORRECT VALUE ***
C        *** OF THE PANEL SUCTION VELOCITIES.                           ***
C
      REWIND 1
      DO I=1,N+1
         WRITE(1,120) XBOD(I),YBOD(I),VN(I)
120      FORMAT(3F10.5)
      END DO
C
C        *** DELETE FILE PARAM.DAT.                                     ***
C
      CLOSE(UNIT=2,STATUS='DELETE')
C
      STOP
      END
```

B.1.1 Sample Input

Below is a listing of sample input data contained in file CASE.DAT.

```
 0.0000        X COORDINATE OF THE JET NOZZLE
 1.0000        X COORDINATE OF THE SHROUD LIP
 0.0000        ROTATION ANGLE OF THE SHROUD LIP (IN DEGREES)
 6.5000        SHROUD LENGTH
 6.5000        X COORDINATE OF THE DIFFUSER START
 0.0000        DIFFUSER SLOPE
 0.0000        FREE STREAM SPEED PARAMETER, GAMMA
15.0000        INITIAL JET CENTERLINE VELOCITY
      1        EXTENDED OUTPUT OPTION (1 FOR EXTRA OUTPUT, 0 FOR STANDARD)
      1        BOUNDARY LAYER CALCULATION CONTROL (1 CALCULATES IT, 0 DOESNT)
 1.00E5        THRUST BASED REYNOLDS NUMBER
```

B.1.2 Sample Output

Below is the output written to file OUT.DAT

```
PATM =    0.79853 PEXIT =    0.65228 R =    0.14625 U10 =   15.02925
PATM =    0.79280 PEXIT =    0.64757 R =    0.14523 U10 =   19.15580
PATM =    1.04161 PEXIT =    1.20321 R =   -0.16160 U10 =   16.98242
PATM =    0.92311 PEXIT =    0.90936 R =    0.01375 U10 =   17.15281
PATM =    0.91671 PEXIT =    0.91357 R =    0.00314 U10 =   17.20317

SEPARATED BOUNDARY LAYER


JET MOMENTUM =    2.72349 EXITING MOMENTUM =    3.91059
INDUCED THRUST COMPUTED FROM SURFACE PRESSURES =    1.25313
INDUCED THRUST COMPUTED FROM MOMENTUM THEOREM  =    1.18709
PRESSURE DRAG ASSOCIATED WITH THE DIFFUSER     =    0.00000
THRUST AUGMENTATION RATIO FROM SURFACE PRESSURES =    1.46012
THRUST AUGMENTATION RATIO FROM MOMENTUM THEOREM  =    1.43587
```

B.2 Dual-Jet Main Program DUOAUG

DUOAUG is the driving program for the dual-jet viscous-inviscid algorithm. Once compiled, it must be linked to the TWINLIB, PAN2LIB, MATHLIB, and IMSL libraries. Input data are to be read from file CASE.DAT.

```
      PROGRAM DUOAUG
C
C************************************************************************
C                                                                       *
C     PROGRAM DUOAUG COMPUTES THE PERFORMANCE OF A TWO-DIMENSIONAL,     *
C INCOMPRESSIBLE FLOW DUAL-JET EJECTOR.  THE CODE IS BASED ON A VISCOUS-    *
C INVISCID ALGORITHM IN WHICH THE INVISCID REGION IS COMPUTED WITH A HIGHER-  *
C ORDER PANEL METHOD AND THE VISCOUS REGION IS COMPUTED WITH AN INTEGRAL     *
C METHOD.                                                               *
C    INPUT DATA IS READ FROM FILE CASE.DAT.  THE ITERATION HISTORY AS WELL AS *
C THE THRUST AUGMENTATION RATIO INFORMATION ARE WRITTEN TO FILE OUT.DAT.  AN  *
C EXTENDED OUTPUT OPTION MAY BE SPECIFIED IN THE INPUT FILE TO CAUSE THE JET  *
C SOLUTION AS WELL AS MORE INFORMATION ABOUT THE MATCHING PROCEDURE TO BE OUT-*
C PUT.                                                                  *
C     THIS PROGRAM MUST BE LINKED TO THE PAN2LIB AND MATHLIB LIBRARIES AS WELL*
C AS THE IMSL LIBRARY MATH LIBRARY.                                     *
C     THIS CODE IS OF EVOLUTIONARY ORIGIN AND CONSEQUENTLY MAY CONTAIN REGIONS*
C OF POOR LOGIC STRUCTURE AND INEFFICIENT PROCEDURES.  THERE HAS BEEN NO    *
C ATTEMPT TO UPGRADE THE CODE TO A "PRODUCTION CODE" STATUS.            *
C                                                                       *
C   *** LATEST REVISION - 23 APR 1987 ***                               *
C                                                                       *
C************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP1
      PARAMETER (MAX=250)
      DIMENSION XBOD(MAX),YBOD(MAX),VN(MAX),XCP(MAX),YCP(MAX),
     1          ALPHA(MAX),D(MAX),ZETA(MAX),CX(3*MAX),CY(3*MAX),
     2          PD(MAX),PE(MAX),PF(MAX),PG(MAX),PH(MAX),PPI(MAX),
     3          C(MAX),IND1(MAX),IND2(MAX),WORK(8*MAX),W(MAX*MAX),
     4          WINV(MAX*MAX),AVEC(MAX),BVEC(MAX),AMAT(MAX*MAX),
     5          BMAT(MAX*MAX),Q(MAX),ALWR(100*MAX),BLWR(100*MAX),
     6          AUPP(100*MAX),BUPP(100*MAX)
C
C        *** AREA21 IS SHARED WITH DERIV2 AND TWOJET                    ***
C
      COMMON U0,U1,P,A,B,Y1,ALP
      COMMON /AREA21/ DIFSLP,XDIFF
C
C        *** OPEN INPUT AND OUTPUT FILES.  CASE.DAT CONTAINS THE INPUT  ***
C        *** VALUES.  OUT.DAT WILL CONTAIN THE OUTPUT.                  ***
C
      OPEN(UNIT= 4,NAME='CASE.DAT'  ,TYPE='OLD',FORM='FORMATTED')
      OPEN(UNIT=21,NAME='OUT.DAT'   ,TYPE='NEW',FORM='FORMATTED')
C
C        *** TOL1 IS THE TOLERANCE FOR THE VISCOUS-INVISCID MATCHING    ***
C        *** TOL2 IS THE TOLERANCE FOT THE EXIT PRESSURE MATCHING       ***
C
      ALP=DLOG(1.0D0+DSQRT(2.0D0))
      TOL1=5.0D-3
      TOL2=1.0D-3
```

```
C
C        *** READ IN THE INPUT VALUES                                       ***
C
      CALL GETPRM(XJ,YJ,Y1DOTO,XEXIT,D1,D2,GAMMA,U10,BO,DUMP1)
      DIFSLP=D1
      XDIFF=D2
C
C        *** WRITE THE FREE STREAM VELOCITY AND THE ANGLE OF ATTACK TO A    ***
C        *** DATA FILE.  THE VALUE OF FREE STREAM SPEED WILL BE CHANGED     ***
C        *** LATER IF REQUIRED.                                             ***
C
      OPEN(UNIT= 2,NAME='PARAM.DAT' ,TYPE='NEW',FORM='FORMATTED')
      V0=0.0D0
      BETA=0.0D0
      WRITE(2,8) V0,BETA
8     FORMAT(2F10.5)
      REWIND 2
C
C        *** WRITE AN INITIAL GUESS FOR THE JET TRAJECTORY TO FILE JETCL.DAT***
C
      OPEN(UNIT=20,NAME='JETCL.DAT' ,TYPE='NEW',FORM='FORMATTED')
      XCL=XJ
      YCL=YJ
      WRITE(20,8) XCL,YCL
      XCL=12.0D0
      YCL=YJ+Y1DOT*(XCL-XJ)
      WRITE(20,8) XCL,YCL
      REWIND 20
C
C        *** IF THE EXTENDED OUTPUT OPTION IS CHOSEN, OPEN ADDITIONAL OUTPUT***
C        *** FILES.  LWRJET.DAT CONTAINS THE VELOCITIES AT THE LOWER SIDE   ***
C        *** OF THE JET.  UPPJET.DAT CONTAINS THE VELOCITIES AT THE UPPER   ***
C        *** SIDE OF THE JET.  MCHJET CONTAINS THE JET SOLUTION OVER THE    ***
C        *** VISCOUS-INVISCID MATCHING REGION.  CHNJET CONTAINS THE JET     ***
C        *** SOLUTION WITHIN THE FULLY VISCOUS REGION.                      ***
C
      IF(DUMP1) THEN
         OPEN(UNIT= 9,NAME='LWRJET.DAT',TYPE='NEW',FORM='FORMATTED')
         OPEN(UNIT=10,NAME='UPPJET.DAT',TYPE='NEW',FORM='FORMATTED')
         OPEN(UNIT=11,NAME='MCHJET.DAT',TYPE='NEW',FORM='FORMATTED')
         OPEN(UNIT=12,NAME='CHNJET.DAT',TYPE='NEW',FORM='FORMATTED')
      END IF
C
10    CONTINUE
C
C        *** OPEN A DATA FILE TO HOLD THE EJECTOR SURFACE COORDINATES       ***
C
      OPEN(UNIT=1,NAME='BODY.DAT',TYPE='NEW',FORM='FORMATTED')
C
C        *** GENERATE THE EJECTOR SURFACE COORDINATES AN THE INITIAL GUESS  ***
C        *** FOR THE PANEL SUCTION VELOCITIES                               ***
C
      CALL DUOBOD(XJ,YJ,DY1DXO,Y1CS,NJLS,NJLF,NJUS,NJUF,NS,NF,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,101)
101      FORMAT(' ERROR IN DUOAUG:  DUOBOD RETURNED WITH IER=1')
         STOP
      END IF
C
C        *** READ THE EJECTOR SURFACE COORDINATES AND THE PANEL SUCTION     ***
C        *** VELOCITIES INTO DATA ARRAYS.                                   ***
```

```
C
      CALL GETDAT(1,XBOD,YBOD,VN,N,VO,BETA)
C
C        *** COMPUTE THE CORRECT VALUE OF THE FREE STREAM SPEED.              ***
C
      CALL FSTRM(U10,BO,1.0DO,0.0DO,GAMMA,VO)
C
C        *** COMPUTE GEOMETRICAL PARAMETERS FOR THE PANEL METHOD.             ***
C
      CALL GEOM(XBOD,YBOD,ZETA,CX,CY,WORK,N,XCP,YCP,ALPHA,D,
     1          IND1,IND2,PD,PE,PF,PG,PH,PPI,C)
C
C        *** COMPUTE THE INFLUENCE COEFFICIENT MATRIX AND ITS INVERSE         ***
C
      CALL INFINV(XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,PF,PG,PH,PPI,C,
     1            WORK,AVEC,BVEC,W,N,AMAT,BMAT,WINV)
C
C        *** COMPUTE THE INFLUENCE VELOCITY COEFFICIENTS ALONG THE JET        ***
C        *** BOUNDARY.                                                        ***
C
      CALL JETMAT(NJLS,NJLF,NJUS,NJUF,XCP,YCP,ALPHA,D,IND1,IND2,
     1            PD,PE,PF,PG,PH,PPI,C,WORK,AVEC,BVEC,AMAT,BMAT,N,
     2            ALWR,BLWR,AUPP,BUPP)
C
C        *** PREPARE FOR AN UPDATED JET TRAJECTORY                            ***
C
      CLOSE(UNIT=20,STATUS='DELETE')
      OPEN(UNIT=20,NAME='JETCL.DAT' ,TYPE='NEW',FORM='FORMATTED')
C
C        *** ENTER A LOOP TO CONVERGE THE EXIT PRESSURE                       ***
C
      JMAX=10
      DO J=1,JMAX
C
C           *** ENTER A LOOP TO PERFORM THE VISCOUS-INVISCID MATCHING         ***
C
         IMAX=10
         DO I=1,IMAX
C
C              *** COMPUTE THE PANEL SOURCE STRENGTHS                         ***
C
            CALL STRNTH(ALPHA,VN,WINV,N,VO,BETA,Q)
C
C              *** COMPUTE THE VELOCITIES ALONG THE JET BOUNDARY              ***
C
            CALL VLCJET(ALWR,BLWR,AUPP,BUPP,Q,NJLS,NJLF,NJUS,NJUF,
     1                  N,VO,BETA,PATM)
C
C              *** COMPUTE THE JET SOLUTION WITHIN THE MATCHING REGION        ***
C
            CALL ONEJET(NJLS,NJLF,NJUS,NJUF,YJ,Y1DOTO,U10,BO,VO,
     1                  ALPHA,VN,N,DUMP1,I,UOO,AO,XEND,Y1END,RES)
C
C              *** COMPUTE THE CORRECT VALUE OF THE FREE STREAM VELOCITY      ***
C
            CALL FSTRM(U10,BO,UOO,AO,GAMMA,VO)
C
C              *** CHECK FOR CONVERGENCE IN THE VISCOUS-INVISCID MATCHING     ***
C
            IF(I.GT.1.AND.DABS(RES).LT.TOL1) GOTO 20
            IF(I.EQ.IMAX) THEN
```

```
              WRITE(3,19)
19            FORMAT(' VISCOUS-INVISCID MATCHING DID NOT CONVERGE ')
              IER=1
              GOTO 200
            END IF
C
         END DO
C
20       CONTINUE
C
C           *** COMPUTE THE JET SOLUTION WITHIN THE FULLY VISCOUS REGION    ***
C
         CALL TWOJET(XEXIT,XEND,DUMP1,PEXIT,DFDRAG,IER)
         IF(IER.EQ.1) GOTO 200
C
C           *** UPDATE THE JET INITIAL VELOCITY                             ***
C
         ROLD=R
         R=(PATM-PEXIT)
         IF(J.EQ.1) THEN
            WW=0.2
         ELSE
            WW=-(U10-U10OLD)/(R-ROLD)
         END IF
         IF(DABS(WW*R).GT.3.0D0) THEN
            WW=WW/2.0D0
         END IF
         U10OLD=U10
         U10=U10+WW*R
C
C           *** WRITE ITERATION INFORMATION                                 ***
C
         WRITE(21,30) PATM,PEXIT,R,U10
30       FORMAT(' PATM = ',F10.5,' PEXIT = ',F10.5,
     1          ' R = ',F10.5,' U10 = ',F10.5)
C
C           *** CHECK TO SEE THAT THE PANEL GEOMETRY IS CONSISTENT WITH     ***
C           *** THE COMPUTED JET CENTERLINE.  IF NOT START OVER BY          ***
C           *** GENERATING A NEW PANEL CONFIGURATION                        ***
C
         DIFF=Y1END-Y1CS
         IF(DABS(R).LT.0.1D0.AND.DABS(DIFF).GT.0.05) THEN
            WRITE(21,25)
25          FORMAT(/,' NEW BODY GENERATED ',/)
            CLOSE(UNIT=1,STATUS='DELETE')
            GOTO 10
         END IF
C
C           *** CHECK FOR CONVERGENCE IN THE EXIT PRESSURE MATCHING         ***
C
         IF(J.GT.1.AND.DABS(R).LT.TOL2) GOTO 40
         IF(J.EQ.JMAX) THEN
            WRITE(3,31)
31          FORMAT(' ERROR IN DUOAUG:  PRESSURE MATCHING DID '
     1             'NOT CONVERGE')
            IER=1
            GOTO 200
         END IF
      END DO
C
C        *** COMPUTE THE THRUST AUGMENTATION RATIO                          ***
```

```
C
40    CALL PERFOR(ALPHA,D,AMAT,BMAT,Q,N,VO,BETA,U10,UOO,AO,BO,
     1            DFDRAG,NS,NF,NJLF,NJUS,PHI)
C
200   CONTINUE
C
C        *** WRITE THE EJECTOR SURFACE COORDINATES ALONG WITH THE CORRECT    ***
C        *** VALUE OF THE PANEL SUCTION VELOCITIES.                          ***
C
      REWIND 1
      DO I=1,N+1
         WRITE(1,111) XBOD(I),YBOD(I),VN(I)
111      FORMAT(3F10.5)
      END DO
C
C        *** CLOSE DATA FILES                                                ***
C
      CLOSE(UNIT=1,STATUS='KEEP')
      CLOSE(UNIT=2,STATUS='KEEP')
      CLOSE(UNIT=4,STATUS='KEEP')
      IF(DUMP1) THEN
         CLOSE(UNIT=9,STATUS='KEEP')
         CLOSE(UNIT=10,STATUS='KEEP')
         CLOSE(UNIT=11,STATUS='KEEP')
         CLOSE(UNIT=12,STATUS='KEEP')
      END IF
      CLOSE(UNIT=20,STATUS='KEEP')
      CLOSE(UNIT=21,STATUS='KEEP')
C
      STOP
      END
```

B.2.1 Sample Input

Below is a set of sample input data contained in file CASE.DAT.

```
  2.0000            X COORDINATE OF THE PRIMARY NOZZLE
  1.0000            Y COORDINATE OF THE PRIMARY NOZZLE
  0.0000            JET INITIAL CENTERLINE SLOPE
 13.2100            X COORDINATE OF SHROUD EXIT
  0.3640            DIFFUSER SLOPE
  9.0000            X COORDINATE OF THE DIFFUSER START
  0.5000            FREE-STREAM SPEED PARAMETER
 10.0000            INITIAL JET CENTERLINE VELOCITY
  0.0150            INITIAL JET HALF-WIDTH
  1                 EXTENDED OUTPUT OPTION: 1 FOR EXTRA OUTPUT 0 FOR PLAIN
```

B.2.2 Sample Output

Below is the output data written to file OUT.DAT

```
PATM =     2.25421 PEXIT =     2.80465 R =    -0.55044 U10 =     9.88991
PATM =     2.28385 PEXIT =     2.81609 R =    -0.53224 U10 =     8.28042
PATM =     1.94640 PEXIT =     2.25369 R =    -0.30729 U10 =     6.08176
PATM =     1.50727 PEXIT =     1.60380 R =    -0.09653 U10 =     5.07471

NEW BODY GENERATED

PATM =     1.28629 PEXIT =     1.31879 R =    -0.03250 U10 =     5.06821
PATM =     1.30357 PEXIT =     1.33681 R =    -0.03323 U10 =     5.36218
PATM =     1.33723 PEXIT =     1.38920 R =    -0.05197 U10 =     4.54686
PATM =     1.26238 PEXIT =     1.26547 R =    -0.00309 U10 =     4.49526
PATM =     1.23384 PEXIT =     1.23263 R =     0.00121 U10 =     4.50974
PATM =     1.22981 PEXIT =     1.22915 R =     0.00066 U10 =     4.52713


SHROUD THRUST SIMPSONS RULE, MIDPOINT RULE:    0.66204    0.66908
NOZZLE CAP THRUST SIMPSONS RULE, MIDPOINT RULE:     0.02953    0.05889


JET MOMENTUM =     0.93306 EXITING MOMENTUM =     1.54414
INDUCED THRUST COMPUTED FROM SURFACE PRESSURES =     0.69157
INDUCED THRUST COMPUTED FROM MOMENTUM THEOREM  =     0.71036
PRESSURE DRAG ASSOCIATED WITH THE DIFFUSER     =     0.22717
THRUST AUGMENTATION RATIO FROM SURFACE PRESSURES =     1.63478
THRUST AUGMENTATION RATIO FROM MOMENTUM THEOREM  =     1.65492
```

B.3 Subroutine Libraries

B.3.1 Single-Jet Library AUGLIB

```
      SUBROUTINE AUGLYR(X,V,N,R,STAG,DUMP,NSTEP,SEP,SCRIT)
C
C**********************************************************************
C     THIS CODE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS    *
C AND ACOUSTICS BY THOMAS LUND.  LATEST REVISION 8 SEPT. 1984.         *
C                                                                      *
C     THIS SUBROUTINE COMPUTES LAMINAR AND TURBULENT BOUNDARY LAYER    *
C DEVELOPMENT, GIVEN AN EXTERNAL VELOCITY DISTRIBUTION. THE EQUATIONS *
C SOLVED HERE ARE BASED ON AN INTEGRAL FORMULATION OF THE BOUNDARY     *
C LAYER EQUATIONS.  IN THE TURBULENT CASE, THE NORMAL TURBULENT        *
C STRESSES ARE NEGLECTED IN COMPARISON WITH THE TURBULENT SHEARING     *
C STRESS.  THE TURBULENT BOUNDARY LAYER EQUATIONS USED HERE ARE FOUND *
C IN SCHLICHTING (7TH ED) P. 676, EQS. (22.7a,b), (22.8a,b), AND       *
C FIG 22.7                                                             *
C     THE VELOCITY DISTRIBUTION DESCRIBED NEED NOT HAVE A              *
C STAGNATION POINT (SEE DESCRIPTION OF PARAMETER STAG).  THE CODE      *
C ASSUMES THAT ALL BOUNDARY LAYERS HAVE A LAMINAR ORIGIN.  TO AVOID    *
C SINGULARITIES AT THE ORIGIN, INITIAL VALUES OF THE VARIOUS CHARAC-   *
C TERISTIC THICKNESSES AND SHAPE FACTORS ARE ASSUMED BY COMPUTING      *
C THESE QUANTITIES AT A SMALL DISTANCE FROM THE ORIGIN USING ANALYTIC *
C EXPRESSIONS FOR A LAMINAR BOUNDARY LAYER IN A ZERO-PRESSURE GRAD-    *
C IENT OUTER STREAM.                                                   *
C     THE LAMINAR BOUNDARY LAYER EQUATIONS ARE MARCHED AWAY FROM THE   *
C INITIAL DATA UNTIL THE END OF THE BODY IS REACHED, OR EITHER TRANS- *
C ITION TO TURBULENT FLOW, OR LAMINAR SEPARATION IS DETECTED.  IF      *
C LAMINAR SEPARATION IS DETECTED, THE CODE HALTS AT THE POINT OF       *
C SEPARATION.  IF TRANSITION IS DETECTED, THE CODE SWITCHES TO THE     *
C TURBULENT BOUNDARY LAYER EQUATIONS, AND CONTINUES TO MARCH UNTIL     *
C EITHER THE END OF THE BODY IS REACHED, OR TURBULENT SEPARATION IS    *
C DETECTED.  IF TURBULENT SEPARATION IS DETECTED, THE CODE HALTS AT    *
C THE POINT OF SEPARATION.                                             *
C     IF OUTPUT IS SPECIFIED (SEE DESCRIPTION OF PARAMETERS DUMP AND   *
C NSTEP) THE FOLLOWING DATA WILL BE PRINTED TO UNIT 3 FOR SPECIFIED    *
C VALUES OF THE SURFACE COORDINATE: SHAPE FACTOR H32, DISPLACEMENT     *
C THICKNESS, MOMENTUM THICKNESS, ENERGY THICKNESS, AND LOCAL SKIN      *
C FRICTION COEFFICIENT                                                 *
C                                                                      *
C **PARAMETER DESCRIPTIONS**                                           *
C                                                                      *
C     INPUT:                                                           *
C         X - VECTOR OF LENGTH N CONTAINING THE VALUES OF THE SURFACE  *
C             COORDINATE AT WHICH EXTERNAL VELOCITIES ARE GIVEN.  THE  *
C             SURFACE COORDINATES MUST START FROM ZERO (X(1)=0.0), BE  *
C             IN INCREASING ORDER, AND BE NORMALIZED BY THE SURFACE    *
C             LENGTH (X(N)=1.0).                                       *
C         V - VECTOR OF LENGTH N CONTAINING THE VALUES OF THE EXTERNAL *
C             VELOCITY WHICH CORRESPOND TO THE SURFACE COORDINATES     *
C             CONTAINED IN VECTOR X.  THE EXTERNAL VELOCITY MUST BE    *
C             NORMALIZED BY THE CHARACTERISTIC VELOCITY OF THE PROBLEM *
C         N - NUMBER OF SURFACE COORDINATE AND EXTERNAL VELOCITY DATA  *
C             PAIRS (LENGTH OF VECTORS X AND V).                       *
C         R - GLOBAL REYNOLDS NUMBER DEFINED AS R=Uc*L/vis, WHERE Uc   *
C             IS THE CHARACTERISTIC VELOCITY OF THE PROBLEM, L IS THE  *
C             SURFACE LENGTH, AND vis IS THE COEFFICIENT OF KINEMATIC  *
C             VISCOSITY.                                               *
C      STAG - LOGICAL VARIABLE USED TO SPECIFY WHETHER OR NOT A        *
```

```
C           STAGNATION POINT EXISTS.  IF STAG IS SET TO .TRUE. A      *
C           STAGNATION POINT IS ASSUMED, IF SET TO .FALSE. NO STAG-   *
C           NATION POINT IS ASSUMED.                                  *
C     DUMP - LOGICAL VARIABLE USED TO SPECIFY WHETHER OR NOT OUTPUT   *
C           IS TO BE GENERATED.  IF DUMP IS SET TO .TRUE. OUTPUT IS   *
C           SENT TO UNIT 3, IF DUMP IS SET TO .FALSE. NO OUTPUT IS    *
C           GENERATED.                                                *
C    NSTEP - INTEGER VALUE USED TO SPECIFY THE NUMBER OF STATIONS AT  *
C           WHICH OUTPUT IS TO BE GENERATED.  THE STATIONS ARE EQUI-  *
C           SPACED.                                                   *
C                                                                     *
C    OUTPUT:                                                          *
C      SEP - LOGICAL VARIABLE USED TO INDICATE A SEPARATED BOUNDARY   *
C           LAYER.  IF EITHER LAMINAR OR TURBULENT SEPARATION IS      *
C           DETECTED, SEP IS SET TO .TRUE.  IF NO SEPARATION IS       *
C           DETECTED, SEP IS SET TO .FALSE.                           *
C   SCRIT  - DIMENSIONLESS SURFACE COORDINATE AT WHICH THE BOUNDARY   *
C           LAYER HAS SEPARATED.  IF NO SEPARATION OCCURS SCRIT = 1   *
C           INDICATING THE END OF THE BODY                            *
C                                                                     *
C**********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      EXTERNAL FCNL,FCNT
      DIMENSION X(100),V(100),C(24),W(2,9),Y(2),YD(2)
      COMMON /BLCVEL/ XX(100),VV(100),RR
      COMMON /BLCSPLN/ SPLN(100),NN
      COMMON /AREA10/ XC
      COMMON /AREA12/ XEXIT
      LOGICAL STAG,LMNR,SEP,DUMP
C
C        *** FUNCTION F1 RETURNS H12 GIVEN H32 ***
C
      F1(H32)=H32/(3.0D0*H32-4.0D0)
C
C        *** FUNCTION WSHR RETURNS THE LOCAL TURBULENT SKIN FRICTION   ***
C        *** COEFFICIENT DIVIDED BY 2, GIVEN THE SHAPE FACTOR H12      ***
C        *** AND THE REYNOLDS NUMBER BASED ON MOMENTUM THICKNESS RD2.  ***
C
      WSHR(H12,RD2)=0.0245D0*(1.0D0-2.0959D0*DLOG10(H12))**1.705D0
     &             /RD2**0.268D0
C
C        *** IN ORDER TO PASS SUBROUTINE ARGUMENTS IN COMMON AS WELL, ***
C        *** WE HAVE TO DEFINE REDUNDANT ARRAYS XX AND VV, AND        ***
C        *** CONSTANTS RR AND NN                                      ***
C
      DO 1 I=1,N
         XX(I)=X(I)
         VV(I)=V(I)
1     CONTINUE
      RR=R
      NN=N
      NF=N-1
C
C        *** SPLINE FIT THE VELOCITY DATA USING AUGLIB ROUTINE LNSPLN ***
C
      CALL LNSPLN(X,V,N,SPLN,IER)
      IF(IER.NE.0) THEN
         WRITE(3,642) IER
642      FORMAT(' IN SUBROUTINE AUGLYR LNSLPLN RETURNED WITH THE ERROR'
     &          'CONDITION IER =',I5)
```

```
      STOP
      END IF
C
C        *** DS IS THE INTEGRATION STEP SIZE, SI IS THE INITIAL CONDITION ***
C        *** STATION                                                      ***
C
      DS=5.0D-4
      SI=0.05D0
C
C      *** DEFINE INTEGRATION DO LOOP UPPER LIMIT ***
C
      IEND=NINT((1.0D0-SI)/DS)
C
C        *** DEFINE THE NUMBER OF INTEGRATION STEPS BETWEEN PRINTOUTS ***
C
      NPRINT=1.0D0/(DFLOAT(NSTEP)*DS)
      CALL LINTRP(SI,X,V,SPLN,N,VI,VID,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,71) SI
71       FORMAT(' IN AUGLYR LINTRP RETURNED WITH AN ERROR FLAG',/,
     &            ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
C
C        *** CHECK FOR STAGNATION POINT, AND SET INITIAL VALUES  ***
C        *** ACCORDINGLY                                         ***
C
      IF(STAG) THEN
         IF(DUMP) WRITE(3,3)
3        FORMAT(10X,' STAGNATION POINT ')
         H32=1.61998D0
         D2=0.29004D0/DSQRT(R*VID)
         D3=H32*D2
      ELSE
         IF(DUMP) WRITE(3,5)
5        FORMAT(10X,' NO STAGNATION POINT ')
         H32=1.57258
         D2=0.66411*DSQRT(SI/(R*VI))
         D3=H32*D2
      END IF
      IF(DUMP) THEN
         WRITE(3,6) R,H32,D2,D3,SI,VI
6        FORMAT(/,10X,' REYNOLDS NUMBER = ',E10.4,//,10X,
     &            ' INITIAL VALUES',//,10X,' H32 = ',E10.4,/,10X,
     &            ' MOMENTUM THICKNESS = ',E10.4,/,10X,
     &            ' ENERGY THICKNESS = ',E10.4,/,10X,' ABSCISSA = ',
     &            E10.4,/,10X,' VELOCITY = ',E10.4,/)
         WRITE(3,7)
7        FORMAT('         X      VELOCITY')
         DO 9 I=1,N
            WRITE(3,8) X(I),V(I)
8           FORMAT(2F10.4)
9        CONTINUE
      END IF
      RD2=R*VI*D2
C
C        *** COMPUTE INITIAL LAMINAR SKIN FRICTION ***
C
      CALL FAPP(H32,H12,EPS,D,KAPS)
      CFL=EPS/RD2
      CD=2.0D0*CFL*VI*VI
```

```
      D1=H12*D2
      IF(DUMP) WRITE(3,10)
10    FORMAT(//,7X,'X',11X,'H32',10X,'D1',11X,'D2',11X,'D3',11X,'CD',/)
      DUM=0.058D0
      IF(DUMP) WRITE(3,20) SI,H32,D1,D2,D3,CD,DUM
20    FORMAT(7E11.4)
C
C        *** INITIALIZE PARAMETERS FOR THE INTEGRATION LOOP ***
C
      LMNR=.TRUE.
      SEP=.FALSE.
      S=SI
      Y(1)=D2
      Y(2)=D3
      RMARGN=1.0D0
      R2=0.058D0
      R3=R2
      R4=R2
      R5=R2
      R6=R2
      NE=2
      TOL=0.001D0
      IND=1
      K=0
C
C        *** ENTER THE INTEGRATION LOOP ***
C
      DO 50 I=1,IEND
         K=K+1
         S=S+DS
C
C           *** INTEGRATE EITHER THE LAMINAR OR TURBULENT BOUNDARY LAYER ***
C           *** EQUATIONS DEPENDING ON THE VALUE OF LMNR USING RK2       ***
C
         IF(LMNR) THEN
            CALL RK2(NE,FCNL,SI,Y,S)
         ELSE
            CALL RK2(NE,FCNT,SI,Y,S)
         END IF
         D2=Y(1)
         D3=Y(2)
         H32=D3/D2
         CALL LINTRP(S,X,V,SPLN,N,VS,VSD,IER)
         IF(IER.EQ.1) THEN
            WRITE(3,72) S
72          FORMAT(' IN AUGLYR LINTRP RETURNED WITH AN ERROR FLAG',/,
     &             ' X HAD THE VALUE',F10.6,' ON ENTRY')
            STOP
         END IF
         RD2=R*VS*D2
C
C           *** IF STILL LAMINAR, CHECK FOR TRANSITION ***
C
         IF(LMNR) THEN
            IF((H32-(DLOG(RD2)+46.78D0)/34.2D0).LE.0.0) THEN
               STRANS=S
               LMNR=.FALSE.
            END IF
         END IF
         IF(LMNR) THEN
C
```

```
C          *** CHECK FOR LAMINAR SEPARATION IGNORE SEPARATION WHICH IS  ***
C          *** PREDICTED DUE TO NOISY VELOCITY DISTRIBUTION BEFORE NOSE ***
C
           IF(H32.LT.1.51509.AND.S.GT.0.7) GOTO 70
C
C             *** COMPUTE LAMINAR SKIN FRICTION ***
C
           CALL FAPP(H32,H12,EPS,D,KAPS)
           CFL=EPS/RD2
           CF=2.0D0*CFL*VS*VS
        ELSE
C
C             *** CHECK FOR TURBULENT SEPARATION ***
C
           IF(H32.LT.1.5) GOTO 70
C
C       *** COMPUTE TURBULENT SKIN FRICTION ***
C
           H12=F1(H32)
           CFT=WSHR(H12,RD2)
           CF=2.0D0*CFT
        END IF
        D1=H12*D2
        IF(K.EQ.NPRINT.AND.DUMP) WRITE(3,20) S,H32,D1,D2,D3,CF,RMARGN
        IF(K.EQ.NPRINT) K=0
50   CONTINUE
     SCRIT=1.0D0
     IF(LMNR) THEN
        IF(DUMP) WRITE(3,60)
60      FORMAT(//,10X,' LAMINAR THROUGHOUT',/10X,' NO SEPARATION')
     ELSE
        IF(DUMP) WRITE(3,65) STRANS
65      FORMAT(//,10X,' TRANSITION AT S = ',F8.4,
    &           /,10X,' NO SEPARATION')
     END IF
     GOTO 200
C
C       *** IF CONTROL IS PASSED TO LINE 70 SEPARATION HAS OCCURRED AND    ***
C       *** THE INTEGRATION IS SUSPENDED AT THE POINT OF SEPARATION.       ***
C
70   SEP=.TRUE.
     SCRIT=S
     IF(LMNR) THEN
        CALL FAPP(H32,H12,EPS,D,KAPS)
        CF=2.0D0*EPS/RD2*VS*VS
        D1=H12*D2
        IF(DUMP) WRITE(3,20) S,H32,D1,D2,D3,CF,RMARGN
        IF(DUMP) WRITE(3,80) S
80      FORMAT(//,10X,' LAMINAR SEPARATION AT S = ',F8.4)
     ELSE
        H12=2.9999D0
        CF=2.0*WSHR(H12,RD2)
        D1=H12*D2
        IF(DUMP) WRITE(3,20) S,H32,D1,D2,D3,CF,RMARGN
        IF(DUMP) WRITE(3,90) STRANS,S
90      FORMAT(//,10X,' TRANSITION AT S = ',F8.4,
    &           /,10X,' TURBULENT SEPARATION AT S = ',F8.4)
     END IF
     GOTO 200
200  RETURN
     END
```

```
      SUBROUTINE BODGEN(XJ,XLIP,THLIP,NJS,NJF,NS,NF)
C
C*****************************************************************************
C                                                                            *
C THIS SUBROUTINE GENERATES AN AUGMENTOR BODY WITH VARIABLE PARAMETERS NOZZLE *
C LOCATION, LIP ROTATION POINT, LIP ROTATION ANGLE, AND MIXING CHAMBER HEIGHT. *
C ON INPUT ALL GEOMETRIC PARAMETERS ARE NORMALIZED BY THE BODY LENGTH.       *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  XJ     - JET NOZZLE POSITION DIVIDED BY THE BODY LENGTH                   *
C  XLIP   - LIP ROTATION POINT DIVIDED BY THE BODY LENGTH                    *
C  THLIP  - LIP ROTATION ANGLE IN RADIANS                                    *
C                                                                            *
C     OUTPUT:                                                                *
C  NJS    - PANEL INDEX OF JET START                                         *
C  NJF    - PANEL INDEX OF JET FINISH                                        *
C  NS     - PANEL INDEX OF BODY NOSE START                                   *
C  NF     - PANEL INDEX OF BODY NOSE FINISH                                  *
C                                                                            *
C  OUTPUT IS PROVIDED IN THE FORM OF DATA FILES.  BODY.DAT CONTAINS THE      *
C  SURFACE COORDINATE PAIRS AS WELL AS THE TRANSPIRATION VELOCITY OVER EACH  *
C  PANEL.                                                                    *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(20),XTEMP(150),YTEMP(150),VNTEMP(150)
      DIMENSION XNOSE(25),YNOSE(25),XSPLN(100),YSPLN(100),SPLN(100,3)
      LOGICAL FLAG
      REWIND 1
      REWIND 2
      PI=3.1415926
5     FORMAT(3F10.5)
C
      THETA=THLIP/180.0D0*PI
C
C        *** DEFINE JET BOUNDARY SLOPE TO BE 12 DEG ***
C
      SLOPE=DTAN(12.0D0/180.0D0*PI)
C
C
C        *** COMPUTE THE CONTROL STATION LOCATION ***
C
      XCONT=XJ+0.7D0/SLOPE
C
      RN=0.5D0
C
C        *** IF THE LIP ROTATION POINT IS LESS THAN THE NOSE RADIUS, SET ***
C        *** THE LIP ROTATION POINT EQUAL TO THE NOSE RADIUS IN ORDER TO ***
C        *** AVOID A CONTORTED BODY SHAPE ***
C
      IF(XLIP.LT.RN) XLIP=RN
C
C        *** CHECK TO INSURE THAT THE CONTROL STATION IS BEHIND THE LIP ***
C        *** ROTATION POINT, IF NOT PRINT ERROR MESSAGE AND SUSPEND     ***
C        *** EXECUTION                                                  ***
C
      IF(XLIP.GT.XCONT) THEN
         WRITE(3,10) XJ,XLIP,THLIP
```

```
10       FORMAT(' IN BODGEN XLIP WAS GREATER THAN XCONT. PARAMETERS',
     &          ' ON ENTRY WERE',/,' XJ =',F8.4,' XLIP =',F8.4,
     &          'THLIP =',F8.4)
         STOP
      END IF
C
C        *** DEFINE EXTREMITIES OF THE SYMMETRY PLANES ***
C
      X1=-20.
      XM=26.
C
C        *** INITIALIZE PARAMETERS ***
C
      FLAG=.TRUE.
      DIST=XJ-X1
      XI=.06
      XIM1=0.
C
C        *** GENERATE A STRING OF COORDINATES WHICH HAVE A RATIO OF ***
C        *** SUCCESSIVE LENGTHS EQUAL TO 1.5 ***
C
      DO 50 I=1,20
         XP(I)=XI
         XI=2.5*XI-1.5*XIM1
         XIM1=XP(I)
         IF(XIM1.GT.DIST) GOTO 60
50    CONTINUE
60    N=I
      Y=0.
      J=0.
      DO 70 I=1,N
         X=XJ-XP(N-I+1)
         J=J+1
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
70    CONTINUE
C
C      *** GENERATE A SET OF COORDINATES FOR THE JET BOUNDARY WHICH HAS THE ***
C      *** FOLLOWING PROPERTIES: PANEL LENGTHS INCREASE IN A RATIO OF 1.5 AS***
C      *** ONE TRAVERSES AWAY FROM THE JET NOZZLE, AND AS ONE TRAVERSES AWAY***
C      *** FROM THE CONTROL STATION MOVING TOWARDS THE NOZZLE.  THE         ***
C      *** INCREASING PANEL LENGTH IS HALTED WHEN THE LENGTH IS             ***
C      *** APPROXIMATELY 0.3.  THE MIDDLE SECTION OF THE JET BOUNDARY HAS   ***
C      *** CONSTANT X INCREMENT OF 0.2851.                                  ***
C
      DX=0.2851D0
      DO 80 I=1,16
         VN=.15*DSQRT(1./(X-XJ+0.1))+.2
         J=J+1
        IF(I.EQ.1) THEN
            NJS=J
            X=XJ
            Y=0.0D0
         END IF
         IF(1.LT.I.AND.I.LE.6) THEN
            X=XJ+XP(I-1)
            Y=SLOPE*(X-XJ)
         END IF
         IF(6.LT.I.AND.I.LE.13) THEN
            X=X+DX
```

```
            Y=SLOPE*(X-XJ)
         END IF
         IF(13.LT.I.AND.I.LE.18) THEN
            X=XCONT-XP(17-I)
            Y=SLOPE*(X-XJ)
         END IF
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
80    CONTINUE
90    NJF=J
C
C        *** GENERATE THE POINTS WHICH DEFINE THE CONTROL STATION ***
C
      X=XCONT
      Y=SLOPE*(X-XJ)
      R=.5*(1.-Y)
      YC=Y+R
      DANG=PI/8.
      ANG=-PI/2.
      DO 100 I=1,8
         VN=DCOS(ANG+DANG/2.)
         J=J+1
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
         ANG=ANG+DANG
         X=XCONT+R*DCOS(ANG)
         Y=YC+R*DSIN(ANG)
100   CONTINUE
C
C        *** GENERATE NOSE POINTS AND STORE ***
C
      XS=RN*(1.-DSIN(THETA))
      X=XS
      Y=1.+DTAN(THETA)*(XLIP-X)
      XC=X+RN*DSIN(THETA)
      YC=Y+RN*DCOS(THETA)
      DEL=0.0
      IF(DABS(DSIN(THETA)).GT.1.E-3)
     &   DEL=2.0D0*RN*(DTAN(THETA)-(1.0D0-DCOS(THETA))/DSIN(THETA))
      ANG=PI
      NRN=NINT(RN*PI/0.15D0)
      DANG=PI/DFLOAT(NRN)
      NCIR=NRN+1
      DO 150 I=1,NCIR
         XNOSE(I)=X
         YNOSE(I)=Y
         ANG=ANG-DANG
         XCI=RN*DCOS(ANG)
         ETA=RN*DSIN(ANG)
         XIM1=X
         X=XC+XCI*DCOS(PI/2.-THETA)-ETA*DSIN(PI/2.-THETA)
         XTMP=X
         Y=YC+XCI*DSIN(PI/2.-THETA)+ETA*DCOS(PI/2.-THETA)
150   CONTINUE
C
C        *** SPLINE FIT THE SECTION BETWEEN THE CONTROL STATION AND NOSE ***
C
      DO 105 I=1,3
         XSPLN(I)=XNOSE(4-I)
```

```
         YSPLN(I)=YNOSE(4-I)
105   CONTINUE
      XSPLN(4)=XLIP
      YSPLN(4)=1.0D0
      DO 107 I=1,3
         XSPLN(4+I)=XCONT-XP(4-I)
         YSPLN(4+I)=1.0D0
107   CONTINUE
      NSPL=7
      NFSPL=6
      CALL ICSCCU(XSPLN,YSPLN,NSPL,SPLN,100,IER)
      IF(IER.EQ.129.OR.IER.EQ.130.OR.IER.EQ.130) THEN
         WRITE(3,109) IER
109      FORMAT(' IN BODGEN ICSCCU RETURNED WITH THE ERROR VALUE ',I5)
         STOP
      END IF
C
C        *** GENERATE POINTS BETWEEN THE CONTROL STATION AND NOSE USING THE ***
C        *** SPLINE FIT                                                     ***
C
      X=XCONT
      Y=1.
      VN=0.0
      J=J+1
      XTEMP(J)=X
      YTEMP(J)=Y
      DO 110 I=1,3
         J=J+1
         X=XCONT-XP(I)
         XTEMP(J)=X
         CALL INTRP(X,XSPLN,YSPLN,NSPL,SPLN,100,Y,YD,YDD,IER)
         YTEMP(J)=Y
         VNTEMP(J)=VN
110   CONTINUE
      DX=0.15D0
      IEND=NINT((X-XS)/DX)
      DX=(X-XS)/DFLOAT(IEND)
      XN=RN*DCOS(THETA)
      FLAG=.TRUE.
      L=0
      DO 120 I=1,IEND-1
         J=J+1
         X=X-DX
         XTEMP(J)=X
         CALL INTRP(X,XSPLN,YSPLN,NSPL,SPLN,100,Y,YD,YDD,IER)
         YTEMP(J)=Y
         VNTEMP(J)=VN
         IF(X.LE.XLIP) THEN
            L=L+1
            IF(L.EQ.1) THEN
               NS=J-1
               FLAG=.FALSE.
            END IF
         END IF
120   CONTINUE
C
C     *** GENERATE THE NOISE POINTS USING THE STORED DATA ***
C
      X=XS
      DO 151 I=1,NCIR
         IF(FLAG.AND.I.EQ.1) NS=J
```

```
            J=J+1
            XTEMP(J)=XNOSE(I)
            YTEMP(J)=YNOSE(I)
            VNTEMP(J)=VN
151     CONTINUE
C
C           *** GENERATE POINTS FROM NOSE TO INFINITY ***
C
        DO 155 I=1,2
            XSPLN(I)=XNOSE(NCIR-2+I)
            YSPLN(I)=YNOSE(NCIR-2+I)
155     CONTINUE
        Y=(1.+2.*RN)+DTAN(THETA)*(XLIP+DEL-XTMP)
        XSPLN(3)=XTMP
        YSPLN(3)=Y
        XSPLN(4)=XLIP+DEL
        YSPLN(4)=1.0D0+2.0D0*RN
        DO 157 I=1,3
            YSPLN(I+4)=1.0D0+2.0D0*RN
157     CONTINUE
        CALL ICSCCU(XSPLN,YSPLN,NSPL,SPLN,100,IER)
        IF(IER.EQ.129.OR.IER.EQ.130.OR.IER.EQ.130) THEN
            WRITE(3,109) IER
            STOP
        END IF
        XI=XTMP
        L=0
160     DO 170 I=1,80
            X=XI
            J=J+1
            IF(X.LT.XCONT-.1) THEN
                CALL INTRP(X,XSPLN,YSPLN,NSPL,SPLN,100,Y,YD,YDD,IER)
            ELSE
                Y=1.0D0+2.0D0*RN
            END IF
            IF(X.GT.(XLIP+DEL)) THEN
                L=L+1
              IF(L.EQ.1) NF=J
            END IF
            XTEMP(J)=X
            YTEMP(J)=Y
            VNTEMP(J)=VN
            XI=2.2*XI-1.2*XIM1
            XIM1=X
            IF(XI.GT.XM) GOTO 220
170     CONTINUE
220     NMAX=J
C
        DO 240 I=1,NMAX
          WRITE(1,5) XTEMP(I),YTEMP(I),VNTEMP(I)
240     CONTINUE
C
        RETURN
        END
```

```
      SUBROUTINE CHANEL(R,XEXIT,XBEGIN,XDIF,DIFSLP,DUMP1,
     1                  PEXIT,DFDRAG)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE CHANEL MARCHES THE JET EQUATIONS FROM THE STATION AT WHICH  *
C THE OUTER VELOCITY HAS BECOME CONSTANT TO THE SHROUD EXIT.  THE INITIAL    *
C CONDITIONS FOR THE TIME MARCH ARE PASSED VIA COMMON BLOCK FROM SUBROUTINE  *
C JET.  SINCE THERE ARE NOW FOUR UNKNOWN QUANTITIES, THE INITIAL CONDITION   *
C VECTOR IS EXTENDED TO 4 ELEMENTS BY INCLUDING AN INITIAL VALUE FOR U0 OF 1.0 *
C                                                                            *
C   *** LATEST REVISION - 25 JAN 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  R      - JET PARAMETERS: U0, U1, P, B, DRAG                               *
C  XEXIT  - X COORDINATE OF THE SHROUD EXIT                                  *
C  XBEGIN - X COORDINATE TO START THE MARCHING                               *
C  XDIF   - X COORDINATE OF THE DIFFUSER START                               *
C  DIFSLP - DIFFUSER SLOPE                                                   *
C  DUMP1  - LOGICAL VARIABLE TO CONTROL OUTPUT                               *
C                                                                            *
C     OUTPUT:                                                                *
C  PEXIT  - PRESSURE AT THE SHROUD EXIT AS COMPUTED BY THE VISCOUS SOLUTION  *
C  DFDRAG - PRESSURE DRAG ASSOCIATED WITH THE DIFFUSER                       *
C  R      - VECTOR CONTAINING THE JET PARAMETERS AT THE SHROUD EXIT          *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP1
      DIMENSION C(24),W(5,9),R(5),RD(5)
      COMMON /DIF/ XD,DS
      EXTERNAL FCN2
C
      XD=XDIF
      DS=DIFSLP
C
      PI=3.14159265D0
      ALP=DLOG(2.0D0)
      N=5
      NW=5
      TOL=1.D-3
      IND=1
C
      U0=R(1)
      U1=R(2)
      P= R(3)
      B= R(4)
      H=1.0D0
      PSTART=P
      HSTART=H
      ETAH=DSQRT(ALP)*H/B
      RMDOT1=B/SQRT(ALP)*(ETAH*U0+DSQRT(PI)/2.0D0*DERF(ETAH)*U1)
      RMJ1=B/SQRT(ALP)*(U0**2*ETAH+DSQRT(PI)*U0*U1*DERF(ETAH)+
     1     0.5D0*DSQRT(PI/2.0D0)*U1**2*DERF(DSQRT(2.0D0)*ETAH))+
     2     P*H
C
C        *** PRINT HEADERS ***
C
```

```
      IF(DUMP1) THEN
         REWIND(12)
         WRITE(12,50) RMJ1,RMDOT1
50       FORMAT(/,25X,' JET IN CHANNEL SOLUTION ',/,
     1         ' INITIAL JET MOMENTUM = ',F10.5,' INITIAL MASS = ',
     2         F10.5)
         WRITE(12,55)
55       FORMAT(/,'          X      U0,U0DOT   U1,U1DOT    P,PDOT',
     1         '      B,BDOT')
      END IF
C
      DX=0.25D0
      DIST=XEXIT-XBEGIN
      NPTS=NINT(DIST/DX)
      DX=DIST/DFLOAT(NPTS)
      X=XBEGIN
C
C        *** MARCH THE VISCOUS SOLUTION ***
C
      DO I=1,NPTS
         XEND=X+DX
         CALL DVERK(M,FCN2,X,R,XEND,TOL,IND,C,MW,W,IER)
C
         IF(DUMP1) THEN
            CALL FCN2(M,X,R,RD)
            WRITE(12,60) X,(R(J),J=1,4),X,(RD(J),J=1,4)
60          FORMAT(5F11.5,/,5F11.5,/)
         END IF
C
      END DO
C
C        *** DEFINE THE EXIT PRESSURE ***
C
      P=R(3)
      H=HSTART+(X-XDIF)*DIFSLP
      PEXIT=P
      DFDRAG=R(5)-(P-PSTART)*HSTART
C
      U0=R(1)
      U1=R(2)
      B= R(4)
      ETAH=DSQRT(ALP)*H/B
      RMDOT=B/SQRT(ALP)*(ETAH*U0+DSQRT(PI)/2.0D0*DERF(ETAH)*U1)
      RMJ=B/SQRT(ALP)*(U0**2*ETAH+DSQRT(PI)*U0*U1*DERF(ETAH)+
     1     0.5D0*DSQRT(PI/2.0D0)*U1**2*DERF(DSQRT(2.0D0)*ETAH))+
     2     P*HSTART+DFDRAG
C
      IF(DUMP1) THEN
         WRITE(12,70) RMJ,RMDOT
70       FORMAT(' FINAL MOMENTUM = ',F10.5, '  FINAL MASS = ',F10.5)
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE FAPP(H32,H12,EPS,D,KAPS)
C
C*****************************************************************************
C                                                                            *
C     THIS SUBROUTINE COMPUTES THE LOCAL SKIN FRICTION COEFFICIENT (EPS), AND *
C THE LOCAL DISSIPATION COEFFICIENT (D) FOR THE LAMINAR BOUNDARY EQUATIONS.  *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  H32  - SHAPE FACTOR                                                       *
C  H12  - SHAPE FACTOR                                                       *
C                                                                            *
C     OUTPUT:                                                                *
C  EPS  - LOCAL SKIN FRICTION COEFFICIENT                                    *
C  D    - LOCAL DISSIPATION COEFFICIENT                                      *
C  KAPS - LAMINAR SEPARATION PARAMETER.  KAPS=1 FOR ATTACHED FLOW AND        *
C         KAPS=0 FOR SEPARATED FLOW                                          *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      KAPS=1
      D=7.853976D0-10.260551D0*H32+3.418898*H32*H32
      IF(H32-1.51509D0)10,20,30
10    KAPS=0
      RETURN
20    H12=4.02922D0-(583.60182D0-724.55916D0*H32+227.1822D0
     &                *H32*H32)*DSQRT(H32-1.51509D0)
      EPS=2.512589D0-1.686095D0*H12+0.391541*H12*H12-0.031729*H12**3.D0
      RETURN
30    IF(H32-1.57258D0)21,21,40
21    GOTO 20
40    H12=79.870845D0-89.582142D0*H32+25.715786D0*H32*H32
      EPS=1.372391-4.226253*H32+2.221687*H32*H32
      RETURN
      END
```

```
      SUBROUTINE FCN1(N,X,S,SD)
C
C*****************************************************************************
C                                                                            *
C     THIS SUBROUTINE COMPUTES THE DERIVATIVES OF THE JET PARAMETERS FOR USE *
C IN MARCHING OF THE VISCOUS SOLUTION WITHIN THE VISCOUS-INVISCID INTERACTION *
C REGION.  THE DERIVATIVE OF U0 IS FOUND FROM THE INVISCID SOLUTION VIA A    *
C LINEAR SPLINE FIT.                                                         *
C                                                                            *
C   *** LATEST REVISION: - 24 JAN 1987 ***                                   *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  N  - NUMBER OF DIFFERENTIAL EQUATIONS IN THE SYSTEM,                      *
C  X  - CARTESIAN COORDINATE                                                 *
C  S  - VECTOR CONTAINING THE VALUES OF U0, U1, P AND B AT THE STATION X     *
C  SD - VECTOR CONTAINING THE DERIVATIVE VALUES OF U0, U1, P AND B AT THE    *
C       STATION X                                                            *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(4),SD(4),C(2,2),WK(2,2),RHS(2)
      COMMON /AREA1/ XE(50),UE(50),SPL(50,3),NJ
C
C        *** FIND THE DERIVATIVE OF U0 THROUGH INTERPOLATION OF THE ***
C        *** SPLINE FIT                                             ***
C
      CALL INTRP(X,XE,UE,NJ,SPL,50,U0,U0DOT,D2,IER)
      IF(IER.NE.0) THEN
         WRITE(3,10) X
10       FORMAT(' IN FCN1 INTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
C
C        *** COMPUTE THE DERIVATIVES OF THE PARAMETERS U0, U1, B, AND P ***
C
      SQRT2=DSQRT(2.0D0)
      ALPHA=DSQRT(DLOG(2.0D0))
      RK=0.0283
C
      U0=S(1)
      U1=S(2)
      P =S(3)
      B =S(4)
C
      C(1,1)=U0+SQRT2*U1
      C(1,2)=U1/B*(U0+SQRT2/2.0D0*U1)
      C(2,1)=SQRT2*U0**2+3.0D0*U0*U1+DSQRT(1.5D0)*U1**2
      C(2,2)=U1/B*(SQRT2*U0**2+1.5D0*U0*U1+
     1             DSQRT(2.0D0/3.0D0)/2.0D0*U1**2)
      D1=2.0D0*U1
      D2=U1*(2.0D0*SQRT2*U0+1.5D0*U1)
      T1=0.0D0
      T2=-RK*(ALPHA**2)*(U1**3)/B
      RHS(1)=T1-D1*U0DOT
      RHS(2)=T2-D2*U0DOT
C
      CALL SIMQ(C,WK,RHS,2,2,IER)
```

```
C
      SD(1)=UODOT
      SD(2)=RHS(1)
      SD(3)=-UO*UODOT
      SD(4)=RHS(2)
C
      RETURN
      END
```

```
      SUBROUTINE FCN2(N,X,R,RD)
C
C******************************************************************************
C                                                                             *
C     THIS SUBROUTINE COMPUTES THE DERIVATIVES OF THE JET PARAMETERS U0, U1,  *
C B, AND P FOR USE IN MARCHING THE VISCOUS SOLUTION IN THE MIXING CHANEL,     *
C DOWNSTREAM OF THE VISCOUS-INVISCID INTERACTION ZONE.                        *
C                                                                             *
C   *** LATEST REVISION - 26 JAN 1987 ***                                     *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C                                                                             *
C     INPUT:                                                                  *
C N  - NUMBER OF DIFFERENTIAL EQUATIONS                                       *
C X  - CARTESIAN COORDINATE                                                   *
C R  - VECTOR CONTAINING THE VALUES OF U0, U1, P, B, AND DRAG AT THE STATION X*
C RD - VECTOR CONTAINING THE DERIVATIVE VALUES OF U0, U1, P, B, AND DRAG      *
C      AT THE STATION X                                                       *
C                                                                             *
C******************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(5),RD(5),A(4,4),T(4),WK(4,4)
      COMMON /DIF/ XDIF,DIFSLP
C
      IF(X.GT.XDIF) THEN
         H=1.0D0+(X-XDIF)*DIFSLP
         HD=DIFSLP
      ELSE
         HD=0.0D0
         H=1.0D0
      END IF
C
      U0=R(1)
      U1=R(2)
      P =R(3)
      B= R(4)
C
      CALL MATRIX(U0,U1,B,H,HD,A,T)
      CALL SIMQ(A,WK,T,4,4,IER)
C
      RD(1)=T(1)
      RD(2)=T(2)
      RD(3)=T(3)
      RD(4)=T(4)
      RD(5)=H*RD(3)
C
      RETURN
      END
```

```
      SUBROUTINE FCNL(NE,S,Y,YD)
C
C*****************************************************************************
C                                                                            *
C     THIS SUBROUTINE COMPUTES THE DERIVATIVES OF D2 AND D3 FOR THE LAMINAR  *
C BOUNDARY LAYER EQUATIONS.  A CALL TO SUBROUTINE FAPP IS NECESSARY.         *
C                                                                            *
C     *** PARAMETER DESCRIPTION ***                                          *
C                                                                            *
C  NE - NUMBER OF DIFFERENTIAL EQUATIONS, IN THIS CASE 2                     *
C  S  - SURFACE COORDINATE                                                   *
C  Y  - VECTOR CONTAINING THE VALUES OF D2 AND D3 AT THE STATION S           *
C  YD - VECTOR CONTAINING THE DERIVATIVE VALUES OF D2 AND D3 AT THE STATION S *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(NE),YD(NE)
      COMMON /BLCVEL/ X(100),V(100),R
      COMMON /BLCSPLN/ SPLN(100),N
      D2=Y(1)
      D3=Y(2)
      H32=D3/D2
C
C        *** COMPUTE THE FRICTION AND DISSIPATION COEFFICIENTS ***
C
      CALL FAPP(H32,H12,EPS,D,KAPS)
C
C        *** COMPUTE THE LOCAL SURFACE VELOCITY AND ITS DERIVATIVE ***
C        *** FROM THE LINEAR SPLINE FIT                             ***
C
      CALL LINTRP(S,X,V,SPLN,N,VS,VSD,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,71) S
71       FORMAT(' IN FCNL LINTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
      RD2=R*VS*D2
      CFL=EPS/RD2
      YD(1)=-(2.0D0+H12)*D2/VS*VSD + CFL
      YD(2)=-3.0D0*D3/VS*VSD + 2.0D0*D/RD2
      RETURN
      END
```

```
      SUBROUTINE FCNT(NE,S,Y,YD)
C
C***************************************************************************
C                                                                          *
C     THIS SUBROUTINE COMPUTES THE DERIVATIVES OF D2 AND D3 FOR USE IN THE *
C MARCHING OF THE TURBULENT BOUNDARY LAYER EQUATIONS.                      *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C     INPUT:                                                               *
C  NE - NUMBER OF DIFFERENTIAL EQUATIONS, IN THIS CASE 2                   *
C  S  - SURFACE COORDINATE                                                 *
C  Y  - VECTOR CONTAINING THE VALUES OF D2 AND D3 AT THE STATION S         *
C  YD - VECTOR CONTAINING THE DERIVATIVE VALUES OF D2 AND D3 AT THE STATION S *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(NE),YD(NE)
      COMMON /BLCVEL/ X(100),V(100),R
      COMMON /BLCSPLN/ SPLN(100),N
C
C        *** FUNCTION F1 RETURNS H12 GIVEN H32 ***
C
      F1(H32)=H32/(3.0D0*H32-4.0D0)
C
C        *** FUNCTION WSHR RETURNS THE LOCAL TURBULENT SKIN FRICTION ***
C        *** COEFFICIENT, GIVEN THE SHAPE FACTOR H12, AND THE REYNOLDS ***
C        *** NUMBER BASED ON MOMENTUM THICKNESS RD2 ***
C
      WSHR(H12,RD2)=0.0245D0*(1.0D0-2.0959D0*DLOG10(H12))**1.705D0
     &              /RD2**0.268D0
C
C        *** FUNCTION CDISS RETURNS THE LOCAL TURBULENT DISSIPATION ***
C        *** COEFFICIENT ***
C
      CDISS(H32,RD2)=(0.00481D0+0.0822D0*(H32-1.5D0)**4.81D0)
     & *(H32/RD3)**(0.2317D0*H32-0.2664D0-0.87D5*(2.0D0-H32)**20)
      D2=Y(1)
      D3=Y(2)
      H32=D3/D2
      H12=F1(H32)
C
C        *** TO AVOID SINGULARITIES AT SEPARATION, PUT BARRIERS ON ***
C        *** H32 AND H12 ***
C
      IF(H32.LT.1.5) H32=1.51
      IF(H12.GT.3.0) H12=2.99
C
C        *** FIND THE LOCAL SURFACE VELOCITY AND ITS DERIVATIVE THROUGH ***
C        *** USE OF THE LINEAR SPLINE FIT ***
C
      CALL LINTRP(S,X,V,SPLN,N,VS,VSD,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,71) S
71       FORMAT(' IN FCNT LINTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
      RD2=R*VS*D2
      RD3=R*VS*D3
```

```
      CT=WSHR(H12,RD2)
      CD=CDISS(H32,RD3)
      YD(1)=-(2.0D0+H12)*D2/VS*VSD + CT
      YD(2)=-3.0D0*Y(2)/VS*VSD + 2.0D0*CD
      RETURN
      END
```

```
      SUBROUTINE FRESTM(U10,BO,UOO,GAMMA,VO)
C
C***************************************************************************
C                                                                          *
C     SUBROUTINE FRESTM COMPUTES THE FREE STREAM VELOCITY WHEN GIVEN THE    *
C PARAMETER GAMMA AND THE PRIMARY JET PARAMETERS.                          *
C                                                                          *
C   *** LATEST REVISION - 23 APRIL 1987 ***                                *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C     INPUT:                                                               *
C U10     - INITIAL JET EXCESS VELOCITY                                    *
C BO      - INITIAL JET HALF-WIDTH                                         *
C UOO     - INITIAL JET EXTERNAL VELOCITY                                  *
C GAMMA  - NON-DIMENSIONAL FREE SPEED PARAMETER                            *
C                                                                          *
C VO      - FREE STREAM VELOCITY                                           *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      ALP=DLOG(2.0DO)
      PI=3.1415926DO
C
C        *** COMPUTE THE PRIMARY JET MOMENTUM FLUX.                        ***
C
      RMJ=DSQRT(PI/ALP)*UOO*U10*BO+0.5DO*DSQRT(PI/2.0DO/ALP)*U10**2*BO
C
C        *** COMPUTE THE FREE STREAM VELOCITY                              ***
C
      VO=GAMMA*DSQRT(RMJ/2.0DO)
C
      RETURN
      END
```

```
      SUBROUTINE GETPRM(XJ,XLIP,THLIP,XEXIT,XDIF,DIFSLP,GAMMA,U10,DUMP1,
     1                  BLAYER,RE)
C
C***********************************************************************
C                                                                      *
C   GETPRM READS PARAMETER VALUES FROM A DATA FILE WHICH IS ASSIGNED UNIT 4  *
C                                                                      *
C   *** LATEST REVISION - 22 APR 1987 ***                              *
C                                                                      *
C   *** PARAMETER DESCRIPTION ***                                      *
C   OUTPUT:                                                            *
C XJ     - X COORDINATE OF THE JET NOZZLE                              *
C XLIP   - X COORDINATE OF THE SHROUD LIP                              *
C THLIP  - SHROUD LIP ROTATION ANGLE (IN DEGREES)                      *
C XEXIT  - X COORDINATE OF THE SHROUD EXIT                             *
C XDIF   - X COORDINATE OF THE DIFFUSER START                          *
C DIFSLP - DIFFUSER SLOPE                                              *
C GAMMA  - FREE-STREAM SPEED PARAMETER                                 *
C U10    - JET INITIAL VELOCITY                                        *
C DUMP1  - OUTPUT CONTROL                                              *
C BLAYER - BOUNDARY LAYER COMPUTATION CONTROL PARAMETER                *
C RE     - REYNOLDS NUMBER BASED ON JET THRUST                         *
C                                                                      *
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP1,BLAYER
C
      READ(4,*) XJ
      READ(4,*) XLIP
      READ(4,*) THLIP
      READ(4,*) XEXIT
      READ(4,*) XDIF
      READ(4,*) DIFSLP
      READ(4,*) GAMMA
      READ(4,*) U10
      READ(4,*) DUMP1
      READ(4,*) BLAYER
      READ(4,*) RE
C
      RETURN
      END
```

```
      SUBROUTINE JET(NJS,NJF,XJET,YJET,UJET,VJET,NJET,U10,BO,VN,N,
     1               DUMP1,NCALL,XEND,R,RES)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE JET PERFORMS THE VISCOUS CALCULATION WITHIN THE VISCOUS-    *
C INVISCID INTERACTION REGION.  THE DERIVATIVE OF U0 IS FOUND FROM THE       *
C INVISCID SOLUTION VIA A QUASI-HERMITE SPLINE FIT, AND IS USED AS A FORCING *
C TERM IN THE VISCOUS SOLUTION.                                              *
C                                                                            *
C   *** LATEST REVISION - 23 APR 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  NJS  - PANEL INDEX OF JET BOUNDARY START                                  *
C  NJF  - PANEL INDEX OF JET BOUNDARY FINISH                                 *
C  XJET - VECTOR OF X COORDINATES ALONG THE JET BOUNDARY                     *
C  YJET - VECTOR OF Y COORDINATES ALONG THE JET BOUNDARY                     *
C  UJET - VECTOR OF HORIZONTAL VELOCITY ALONG THE JET BOUNDARY               *
C  VJET - VECTOR OF VERTICAL VELOCITY ALONG THE JET BOUNDARY                 *
C  NJET - NUMBER OF POINTS ALONG THE JET BOUNDARY                            *
C  U10  - JET INITIAL CENTERLINE VELOCITY                                    *
C  BO   - JET INITIAL VELOCITY HALF-WIDTH                                    *
C  VN   - VECTOR CONTAINING THE NORMAL VELOCITIES TO THE PANELS ALONG THE JET *
C         BOUNDARY IN THE VISCOUS-INVISCID INTERACTION REGION                *
C  N    - NUMBER OF PANELS                                                   *
C  DUMP1 -LOGICAL VARIABLE FOR OUTPUT CONTROL                                *
C  NCALL -INDEX TO KEEP TRACK OF THE SUCCESSIVE CALLS TO JET                 *
C                                                                            *
C     OUTPUT:                                                                *
C  VN   - UPDATED NORMAL VELOCITY VECTOR                                     *
C  XEND - X STATION AT WHICH THE VISCOUS-INVISCID MATCHING ENDS              *
C  R    - VECTOR CONTAINING THE VALUES OF THE JET PARAMETERS AT THE END OF   *
C         THE VISCOUS-INVISCID MATCHING REGION                               *
C  RES  - MAXIMUM RESIDUAL IN THE VISCOUS-INVISCID MATCHING                  *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP1
      DIMENSION XJET(NJET),YJET(NJET),UJET(NJET),VJET(NJET),VN(N),
     1          W(4,9),C(24),S(4),SD(4),SPLN(50,3),R(5)
      DIMENSION XTMP(300),YTMP(300)
      COMMON /AREA1/ XE(50),UE(50),SPL(50,3),NJ
      EXTERNAL FCN1
C
      PI=3.141592D0
      ALP=DLOG(2.0D0)
      M=4
      MW=4
      TOL=1.D-4
      IND=1
C
C        *** PRINT HEADERS ***
C
      IF(DUMP1) THEN
         REWIND 9
         REWIND 10
         WRITE(9,45)
45       FORMAT(/,25X, ' JET VELOCITIES ')
```

```
         WRITE(9,40)
40       FORMAT(/,'      X          Y        UINV      VINV      VVIS',
     &         '     VNOLD      VNNEW      RES')
         WRITE(10,50)
50       FORMAT(/,25X,' JET SOLUTION ')
         WRITE(10,55)
55       FORMAT(/,'      X    U0,U0DOT  U1,U1DOT   P,PDOT',
     1         '    B,BDOT')
      END IF
C
C        *** SPLINE FIT THE HORIZONTAL COMPONENT OF INVISCID VELOCITY ***
C        *** ALONG THE JET BOUNDARY                                   ***
C
      CALL IQHSCU(XJET,UJET,NJET,SPLN,50,IER)
      IF(IER.NE.0) THEN
         WRITE(3,90) IER
90       FORMAT('AFTER CALL TO SPLINE IER HAS THE ERROR VALUE ',I5)
         STOP
      END IF
C
C        *** DUPLICATE THE INVISCID VELOCITY DATA SO THAT IT MAY BE ***
C        *** SENT IN COMMON                                         ***
C
      NJ=NJET
      DO I=1,NJET
         XE(I)=XJET(I)
         UE(I)=UJET(I)
         SPL(I,1)=SPLN(I,1)
         SPL(I,2)=SPLN(I,2)
         SPL(I,3)=SPLN(I,3)
      END DO
C
C        *** DEFINE INITIAL VALUES OF THE JET PARAMETERS ***
C        *** S(1)<--U0, S(2)<--U1, S(3)<--P, S(4)<--B    ***
C
      U00=UJET(2)
      P0=0.0D0
C
      S(1)=U00
      S(2)=U10
      S(3)=P0
      S(4)=B0
C
      RES=0.0
C
C        *** ENTER LOOP TO MARCH THE VISCOUS EQUATIONS
C
      X=XJET(2)-.001D0
      DO 10 J=2,NJET
         XEND=XJET(J)
         CALL DVERK(M,FCN1,X,S,XEND,TOL,IND,C,MW,W,IER)
         IF(IND.LT.0.OR.IER.GT.0) THEN
            WRITE(3,150) IND,IER
150         FORMAT(/,'IN JET IND= ',I5,'  IER= ',I5,/)
            STOP
         END IF
C
C           *** OBTAIN THE LOCAL DERIVATIVE VALUES OF THE JET PARAMETERS ***
C
   CALL FCN1(M,XEND,S,SD)
C
```

```
C          *** COMPUTE THE LOCAL INVISCID VELOCITY AND ITS DERIVATIVE ***
C
        CALL INTRP(XEND,XJET,UJET,NJET,SPLN,50,U0,U0DOT,D2,IER)
        IF(IER.EQ.1) THEN
           WRITE(3,12) XEND
12         FORMAT(' IN JET INTRP RETURNED WITH AN ERROR FLAG',/,
     &            ' X HAD THE VALUE',F10.6,' ON ENTRY')
           STOP
        END IF
C
C          *** COMPUTE THE VERTICAL COMPONENT OF VELOCITY AT THE JET ***
C          *** BOUNDARY FROM THE VISCOUS SOLUTION                    ***
C
        VVIS=V(S,SD,YJET(J))
C
C          *** COMPUTE THE LOCAL RESIDUAL BY COMPARING THE VISCOUS AND ***
C          *** INVISCID VERTICAL COMPONENTS OF VELOCITY ALONG THE JET  ***
C          *** BOUNDARY                                                 ***
C
        RR=VVIS-VJET(J)
        IF(DABS(RR).GT.RES) RES=DABS(RR)
C
C          *** MAKE A CORRECTION TO THE LOCAL ENTRAINMENT VELOCITY ***
C
        W1=1.0D0-0.7D0/DFLOAT(NJET-2)*DFLOAT(J-2)

        JJ=NJS-1+J
        VNEW= VN(JJ)-W1*RR
C
        IF(DUMP1) THEN
           WRITE(9,60) XJET(J),YJET(J),UJET(J),VJET(J),VVIS,VN(JJ),
     1                 VNEW,RR
60         FORMAT(8F10.5)
           WRITE(10,65) XJET(J),S(1),S(2),S(3),S(4),
     1                  XJET(J),SD(1),SD(2),SD(3),SD(4)
65         FORMAT(5F10.5,/,5F10.5,/)
        END IF
C
C          *** MAKE FIRST PANEL SUCTION EQUAL TO THE SECOND TO ENHANCE ***
C          *** STABILITY                                                ***
C
        IF(J.EQ.2) VN(NJS)=VNEW
        VN(NJS-1+J)=VNEW
9       CONTINUE
10    CONTINUE
C
      REWIND 1
      DO I=1,N
         READ(1,*) XTMP(I),YTMP(I)
      END DO
      REWIND 1
      DO I=1,N
         WRITE(1,11) XTMP(I),YTMP(I),VN(I)
11       FORMAT(3F10.5)
      END DO
C
C        *** INITIALIZE PARAMETERS FOR THE CHANNEL SOLUTION ***
C
      R(1)=S(1)
      R(2)=S(2)
      R(3)=S(3)
```

```
      R(4)=S(4)
      R(5)=0.0D0
C
      RETURN
      END
```

```
      SUBROUTINE JETCOF(NJS,NJF,XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,PF,
     1                  PG,PH,PPI,C,WORK,A,B,AMAT,BMAT,N,
     2                  XJET,YJET,NJET,AJET,BJET)
C
C***********************************************************************
C                                                                      *
C   SUBROUTINE JETCOF COMPUTES THE INFLUENCE COEFFICIENTS FOR THE MATCHING  *
C POINTS ALONG THE JET BOUNDARY.                                       *
C                                                                      *
C   LATEST REVISION 23 APR  1987                                       *
C                                                                      *
C   *** PARAMETER DESCRIPTION ***                                      *
C                                                                      *
C     INPUT:                                                           *
C  NJS   - PANEL NUMBER OF THE BEGINNING OF THE JET BOUNDARY           *
C  NJF   - PANEL NUMBER OF THE END OF THE JET BOUNDARY                 *
C  XCP   - VECTOR OF CONTROL POINT X COORDINATES                       *
C  YCP   - VECTOR OF CONTROL POINT Y COORDINATES                       *
C  ALPHA - VECTOR CONTAINING THE SURFACE SLOPES                        *
C  D     - VECTOR CONTAINING THE PANEL LENGTHS                         *
C  IND1  - VECTOR OF INDEX OF PANEL ADJOINING TO THE LEFT              *
C  IND2  - VECTOR OF INDEX OF PANEL ADJOINING TO THE RIGHT             *
C PD..PPI- SOURCE PARABOLIC FIT COEFFICIENTS                           *
C  C     - VECTOR OF SURFACE CURVATURE COEFFICIENTS                    *
C  WORK  - WORK SPACE VECTOR                                           *
C  A     - WORK SPACE VECTOR                                           *
C  B     - WORK SPACE VECTOR                                           *
C  AMAT  - MATRIX OF X COMPONENT INDUCED VELOCITIES                    *
C  BMAT  - MATRIX OF Y COMPONENT INDUCED VELOCITIES                    *
C  N     - NUMBER OF PANELS                                            *
C                                                                      *
C     OUTPUT:                                                          *
C  XJET  - VECTOR OF X COORDINATES OF THE CONTROL POINTS ALONG THE BOUNDARY  *
C  YJET  - VECTOR OF Y COORDINATES OF THE CONTROL POINTS ALONG THE BOUNDARY  *
C  NJET  - NUMBER OF POINTS ALONG THE JET BOUNDARY                     *
C  AJET  - MATRIX OF U-VELOCITY INFLUENCE COEFFICIENTS FOR THE JET BOUNDARY  *
C  BJET  - MATRIX OF V-VELOCITY INFLUENCE COEFFICIENTS FOR THE JET BOUNDARY  *
C                                                                      *
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XCP(N),YCP(N),ALPHA(N),D(N),PD(N),PE(N),PF(N),PG(N),
     1          PH(N),PPI(N),C(N),IND1(N),IND2(N),WORK(8*N),A(N),B(N),
     2          XJET(NJET),YJET(NJET),AJET(NJET,N),BJET(NJET,N),
     3          AMAT(N,N),BMAT(N,N)
C
C
C        *** CALCULATE AND STORE THE INFLUENCE COEFFICIENTS FOR THE ***
C        *** JET BOUNDARY                                           ***
C
      DO I=NJS,NJF
         II=I-NJS+1
         X=XCP(I)
         Y=YCP(I)
         XJET(II)=X
         YJET(II)=Y
C
         DO J=1,N
            AJET(II,J)=AMAT(I,J)
            BJET(II,J)=BMAT(I,J)
         END DO
```

```
C
C          *** COMPUTE THE VELOCITY AT A POINT SLIGHTLY ABOVE THE JET      ***
C          *** BOUNDARY WHEN NEAR THE CONTROL STATION TO AVOID THE SPIKE   ***
C          *** IN THE VELOCITY FIELD CAUSED BY THE CURVATURE DISCONTINUITY ***
C          *** AT THE CONTROL STATION.                                     ***
C
         IF(I.GT.(NJF-5)) THEN
            YM=Y+(X-XCP(NJF-5))*0.15D0
            CALL INFLCE(X,YM,XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,
     1                 PF,PG,PH,PPI,C,WORK,N,A,B)
            DO J=1,N
               AJET(II,J)=A(J)
            END DO
         END IF
C
      END DO
C
      RETURN
      END
```

```
      SUBROUTINE JETVEL(AJET,BJET,NJET,Q,N,V0,BETA,UJET,VJET,U00,PATM)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE VLCJET COMPUTES VALUES OF THE VELOCITY COMPONENTS AT THE    *
C  JET BOUNDARY                                                              *
C                                                                            *
C    LATEST REVISION 23 APR 1986                                             *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  AJET  - INFLUENCE COEFFICIENTS FOR U-VELOCITY ALONG THE JET BOUNDARY      *
C  BJET  - INFLUENCE COEFFICIENTS FOR V-VELOCITY ALONG THE JET BOUNDARY      *
C  NJET  - NUMBER OF PANELS ALONG THE JET BOUNDARY                           *
C  Q     - VECTOR OF SOURCE STRENGTHS                                        *
C  N     - NUMBER OF PANELS                                                  *
C  V0    - FREE STREAM SPEED                                                 *
C  BETA  - ANGLE OF ATTACK                                                   *
C                                                                            *
C     OUTPUT:                                                                *
C  UJET  - VECTOR OF HORIZONTAL COMPONENT OF VELOCITY ALONG THE JET BOUNDARY *
C  VJET  - VECTOR OF VERTICAL COMPONENT OF VELOCITY ALONG THE JET BOUNDARY   *
C  U00   - U0 COMPONENT OF VELOCITY AT THE JET NOZZLE                        *
C  PATM  - UPSTREAM AMBIENT PRESSURE                                         *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AJET(NJET,N),BJET(NJET,N),Q(N),UJET(NJET),VJET(NJET)
C
C        *** COMPUTE THE VELOCITY COMPONENTS ALONG THE JET BOUNDARY        ***
C
      DO I=1,NJET
         SUM1=0.0D0
         SUM2=0.0D0
         DO J=1,N
            SUM1=SUM1+AJET(I,J)*Q(J)
            SUM2=SUM2+BJET(I,J)*Q(J)
         END DO
         UJET(I)=V0*DCOS(BETA)+SUM1
         VJET(I)=V0*DSIN(BETA)+SUM2
      END DO
C
C        *** CALCULATE THE UPSTREAM ATMOSPHERIC PRESSURE                   ***
C
      PATM=0.5D0*(UJET(2)**2-V0**2)
C
      U00=UJET(2)
C
      RETURN
      END
```

```
      SUBROUTINE MATRIX(U0,U1,B,H,HDOT,A,T)
C
C*****************************************************************************
C                                                                            *
C  SUBROUTINE MATRIX COMPUTES THE MATRIX ELEMENTS AND RIGHT HAND SIDE OF THE *
C EQUATIONS FOR THE DERIVATIVES OF THE JET PARAMETERS.                       *
C                                                                            *
C   *** LATEST REVISION - 26 JAN 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C     INPUT:                                                                 *
C  U0      - JET EXTERNAL VELOCITY                                           *
C  U1      - JET CENTERLINE EXCESS VELOCITY                                  *
C  B       - JET EXCESS VELOCITY HALF-WIDTH                                  *
C  H       - CHANNEL HALF-WIDTH                                              *
C  HDOT    - CHANNEL SLOPE                                                   *
C                                                                            *
C     OUTPUT:                                                                *
C  A       - MATRIX ELEMENTS                                                 *
C  T       - RIGHT HAND SIDE VECTOR                                          *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4),T(4)
C
      ALP=DSQRT(DLOG(2.0D0))
      PI=3.14159265D0
C
      ETAH=ALP*H/B
      ETAH2=ETAH**2
      ETAH3=ETAH**3
      E1=DEXP(-ETAH2)
      E2=DEXP(-2.0D0*ETAH2)
      F1=DSQRT(PI)/2.0D0*DERF(ETAH)
      F2=DSQRT(PI/2.0D0)/2.0D0*DERF(DSQRT(2.0D0)*ETAH)
      UH=U0+U1*E1
      AUX1=U0-UH/2.0D0
      AUX2=F1-ETAH*E1
      AUX3=F2-ETAH*E2
      AUX4=1.0D0-E1
      AUX5=1.0D0-E2
      AUX6=(2.0D0*AUX2-F1)*F1
      AUX7=(1.0D0+ETAH2)*E1
      RK=0.0283D0
      C5=2.0D0*RK*(ALP**2)*(U1**2)/B
C
      A(1,1)=AUX1*ETAH+U1*F1
      A(1,2)=AUX1*F1+U1*F2
      A(1,3)=0.5D0*ETAH
      A(1,4)=(U1/B)*(AUX1*AUX2+0.5D0*U1*AUX3)
      A(2,1)=U0*ETAH2+U1*(3.0D0*AUX4-2.0D0*ETAH2*E1)
      A(2,2)=U0*AUX4+U1*(AUX5+AUX6)
      A(2,3)=ETAH2
      A(2,4)=(U1/B)*(2.0D0*U0*(1.0D0-AUX7)+U1*(AUX6+0.5D0*AUX5))
      A(3,1)=1.0D0/3.0D0*U0*ETAH3+U1*(2.0D0*AUX2-ETAH3*E1)
      A(3,2)=0.5D0*U0*AUX2+U1*(0.5D0*AUX3+F2-AUX7*F1)
      A(3,3)=1.0D0/3.0D0*ETAH3
      A(3,4)=(U1/B)*(U0*(1.5D0*AUX2-ETAH3*E1)+
     1                  U1*(-AUX7*F1+F2+0.25D0*AUX3))
      A(4,1)=ETAH
```

```
      A(4,2)=F1
      A(4,3)=0.0D0
      A(4,4)=(U1/B)*AUX2
C
      T(1)=0.0D0
      T(2)=C5*AUX4
      T(3)=C5*AUX2
      T(4)=-(ALP/B)*UH*HDOT
C
      RETURN
      END
```

```
      SUBROUTINE PERFRM(R,HEXIT,ALPHA,D,AMAT,BMAT,Q,N,VO,BETA,
     1                  U10,UOO,BO,DFDRAG,NS,NF,PHI)
C
C***************************************************************************
C                                                                          *
C     THIS SUBROUTINE COMPUTES THE THRUST AUGMENTATION RATIO IN TWO        *
C INDEPENDENT CALCULATIONS; BY INTEGRATION OF THE SURFACE PRESSURES, AND BY A *
C CONTROL VOLUME ANALYSIS USING THE BLASIUS MOMENTUM THEOREM.  A SUMMARY OF  *
C THE PERFORMANCE PARAMETERS ARE WRITTEN TO THE OUTPUT FILE OUT.DAT.        *
C                                                                          *
C   *** LATEST REVISION - 23 APR 1987 ***                                  *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C     INPUT:                                                               *
C  R      - VECTOR CONTAINING THE JET PARAMETERS AT THE SHROUD EXIT        *
C  HEXIT  - CHANNEL EXIT HALF WIDTH                                        *
C  ALPHA  - VECTOR OF PANEL ORIENTATION ANGLES                             *
C  D      - VECTOR OF PANEL LENGTHS                                        *
C  AMAT   - MATRIX OF X COMPONENT INDUCED VELOCITIES                       *
C  BMAT   - MATRIX OF Y COMPONENT INDUCED VELOCITIES                       *
C  Q      - VECTOR CONTAINING THE SOURCE STRENGTHS                         *
C  N      - NUMBER OF PANELS                                               *
C  VO     - FREE-STREAM SPEED                                              *
C  BETA   - ANGLE OF ATTACK                                                *
C  U10    - JET INITIAL CENTERLINE VELOCITY                                *
C  UOO    - INITIAL UO COMPONENT OF VELOCITY                               *
C  BO     - INITIAL JET VELOCITY HALF-WIDTH                                *
C  DFDRAG- PRESSURE DRAG ASSOCIATED WITH THE DIFFUSER                      *
C  NS     - PANEL INDEX OF THE SHROUD NOSE START                           *
C  NF     - PANEL INDEX OF THE SHROUD NOSE FINISH                          *
C                                                                          *
C     OUTPUT:                                                              *
C  PHI    - THRUST AUGMENTATION AS COMPUTED BY THE MOMENTUM THEOREM        *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(5),ALPHA(N),D(N),AMAT(N,N),BMAT(N,N),Q(N)
C
      PI=3.14159265DO
      ALP=DSQRT(DLOG(2.ODO))
C
C         *** COMPUTE THE PRIMARY JET MOMENTUM FLUX                         ***
C
      RMJ=BO/ALP*DSQRT(PI)*(UOO*U10+0.5DO/DSQRT(2.ODO)*U10**2-
     1                      0.5DO*VO*U10)
C
C         *** INTEGRATE THE SURFACE PRESSURES                               ***
C
      SUM3=0.ODO
      VOX=VO*DCOS(BETA)
      VOY=VO*DSIN(BETA)
C
      DO I=NS,NF
C
         SUM1=0.ODO
         SUM2=0.ODO
         DO J=1,N
            SUM1=SUM1+AMAT(I,J)*Q(J)
            SUM2=SUM2+BMAT(I,J)*Q(J)
```

```
          END DO
C
          U=SUM1+VOX
          V=SUM2+VOY
C
          SUM3=SUM3+0.5D0*(U*U+V*V-V0**2)*D(I)*DSIN(ALPHA(I))
C
       END DO
C
C         *** TAUX IS THE INDUCED THRUST                                 ***
C
       TAUX=SUM3
C
       U0=R(1)
       U1=R(2)
       P =R(3)
       B =R(4)
       ETAH=ALP*HEXIT/B
C
C         *** COMPUTE THE MOMENTUM FLUX EXITING FROM THE EJECTOR          ***
C
       TGROSS=B/ALP*(U0**2*ETAH+DSQRT(PI)*U0*U1*DERF(ETAH)+
     1              0.5D0*DSQRT(PI/2.0D0)*U1**2*
     2              DERF(DSQRT(2.0D0)*ETAH))-
     3       V0*B/ALP*(ETAH*U0+DSQRT(PI)/2.0D0*U1*DERF(ETAH))
C
C         *** COMPUTE THE THRUST AUGMENTATION RATION USING THE            ***
C         *** MOMENTUM THEOREM AND SURFACE PRESSURE CALCULATION           ***
C
       PHIMT=TGROSS/RMJ
       PHISP=1.0D0+(TAUX-DFDRAG)/RMJ
C
C         *** COMPUTE THE INDUCED THRUST FROM THE MOMENTUM THEOREM        ***
C
       TIND=TGROSS-RMJ+DFDRAG
C
C         *** WRITE RESULTS TO FILE OUT.DAT.                              ***
C
       WRITE(21,10) RMJ,TGROSS,TAUX,TIND,DFDRAG,PHISP,PHIMT
10     FORMAT(//,' JET MOMENTUM = ',F10.5,
     1          ' EXITING MOMENTUM = ',F10.5,/,
     2 ' INDUCED THRUST COMPUTED FROM SURFACE PRESSURES = ',F10.5,/,
     3 ' INDUCED THRUST COMPUTED FROM MOMENTUM THEOREM  = ',F10.5,/,
     4 ' PRESSURE DRAG ASSOCIATED WITH THE DIFFUSER     = ',F10.5,/,
     4 ' THRUST AUGMENTATION RATIO FROM SURFACE PRESSURES = ',F10.5,/,
     5 ' THRUST AUGMENTATION RATIO FROM MOMENTUM THEOREM  = ',F10.5)
C
C         *** CHOOSE THE MOMENTUM THEOREM CALCULATED VALUE OF PHI         ***
C
       PHI=PHIMT
C
       RETURN
       END
```

C-3

```
      SUBROUTINE SURFVEL(XCONT,XEXIT,XCP,YCP,D,AMAT,BMAT,Q,N,
     1                   V0,BETA,SC,UEXT,NEXT,XLEN,STAG)
C
C***************************************************************************
C                                                                          *
C    THIS SUBROUTINE COMPUTES THE SHROUD SURFACE VELOCITY FROM THE INVISCID *
C SOLUTION FOR USE IN THE BOUNDARY LAYER CALCULATION.                      *
C                                                                          *
C   *** LATEST REVISION - 22 APRIL 1987 ***                                *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C    INPUT:                                                                *
C  XCONT - X COORDINATE OF THE CONTROL STATION                             *
C  XEXIT - X COORDINATE OF THE SHROUD EXIT                                 *
C  XCP   - VECTOR CONTAINING THE X COORDINATES OF THE CONTROL POINTS       *
C  YCP   - VECTOR CONTAINING THE Y COORDINATES OF THE CONTROL POINTS       *
C  D     - VECTOR CONTAINING THE PANEL LENGTHS                             *
C  AMAT  - MATRIX OF HORIZONTAL INDUCED VELOCITIES                         *
C  BMAT  - MATRIX OF VERTICAL INDUCED VELOCITIES                           *
C  Q     - VECTOR CONTAINING THE SOURCE STRENGTHS                          *
C  N     - NUMBER OF PANELS                                                *
C  V0    - FREE-STREAM SPEED                                               *
C  BETA  - ANGLE OF ATTACK                                                 *
C  SC    - VECTOR OF SURFACE COORDINATES AT WHICH THE VELOCITIES ARE       *
C          CALCULATED.  THE SURFACE COORDINATES ARE NORMALIZED SUCH THAT THE *
C          CONTROL STATION LOCATION IS 1.  THE ORIGIN IS THE STAGNATION POINT *
C          IF A FREE-STREAM IS PRESENT AND THE SHROUD TRAILING EDGE FOR    *
C          STATIC OPERATION                                                *
C  UEXT  - VECTOR CONTAINING THE SURFACE VELOCITIES                        *
C  NEXT  - NUMBER OF STATIONS AT WHICH THE VELOCITY IS CALCULATED          *
C  XLEN  - LENGTH OF THE SURFACE OVER WHICH THE THE VELOCITIES ARE CALCULATED *
C  STAG  - LOGICAL VARIABLE SET TO TRUE WHEN A STAGNATION POINT IS PRESENT *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL STAG
      DIMENSION XCP(N),YCP(N),D(N),AMAT(N,N),BMAT(N,N),Q(N)
      DIMENSION SC(100),UEXT(100)
      LOGICAL FLAG
C
      V0X=V0*DCOS(BETA)
      V0Y=V0*DSIN(BETA)
C
C        *** FIND PANEL INDEX OF SHROUD TRAILING EDGE ***
C
      DO 10 I=N,1, -1
         IF(XCP(I-1).LT.XEXIT) GOTO 20
10    CONTINUE
20    NS=I
      NSJ=NS
C
C        *** FIND THE PANEL INDEX OF THE CONTROL STATION ***
C
      FLAG=.FALSE.
      DO 30 I=NS,1, -1
         IF(XCP(I-1).GT.XCP(I)) FLAG=.TRUE.
         IF(FLAG.AND.XCP(I).GT.XCONT) GOTO 40
30    CONTINUE
40    NF=I+1
```

```
      NFJ=I
      K=0
C
C        *** STORE THE SURFACE COORDINATES AND COMPUTE THE ***
C        *** SURFACE VELOCITIES                            ***
C
      DO 100 I=NS,NF,-1
         IF(I.EQ.NS) THEN
            K=K+1
         S=XEXIT-XCP(I)
            SC(1)=S
            X=XCP(I)
            Y=XCP(I)
            SUM1=0.0D0
            SUM2=0.0D0
            DO J=1,N
               SUM1=SUM1+AMAT(I,J)*Q(J)
               SUM2=SUM2+BMAT(I,J)*Q(J)
            END DO
            U=SUM1+VOX
            V=SUM2+VOY
            UEXT(K)=DSQRT(U*U+V*V)
         ELSE
            S=S+D(I+1)/2.0D0+D(I)/2.0D0
C
C              *** FILTER THE VELOCITY DATA WHICH IS TAKEN IN A REGION ***
C              *** ADJACENT TO THE CONTROL STATION SINGULARITY.        ***
C
            X=XCP(I)
            Y=XCP(I)
            SUM1=0.0D0
            SUM2=0.0D0
            DO J=1,N
               SUM1=SUM1+AMAT(I,J)*Q(J)
               SUM2=SUM2+BMAT(I,J)*Q(J)
            END DO
            U=SUM1+VOX
            V=SUM2+VOY
            UMOD=DSQRT(U*U+V*V)
            IF(S.LT.5.0) THEN
C
C                 *** INCLUDE THE LOCAL POINT ONLY IF THE ***
C                 *** VELOCITY IS INCREASING              ***
C
            IF(UMOD.GT.UEXT(K)) THEN
               K=K+1
               SC(K)=S
               UEXT(K)=UMOD
            END IF
            ELSE
               K=K+1
               SC(K)=S
               UEXT(K)=UMOD
            END IF
         END IF
100   CONTINUE
C
C        *** SEARCH FOR THE STAGNATION POINT (MINIMUM VELOCITY MODULUS) ***
C
      UMIN=10.0D0
      DO 105, I=1,K
```

```
         IF(UEXT(I).LT.UMIN) THEN
            UMIN=UEXT(I)
            L=I
         END IF
105   CONTINUE
      IF(L.EQ.1) THEN
         STAG=.FALSE.
      ELSE
         STAG=.TRUE.
      END IF
C
C        *** CORRECT IF NOT ALL DATA IS FROM THE SAME SIDE OF THE ***
C        *** STAGNATION POINT                                     ***
C
      IF(STAG) THEN
         TEST=(UEXT(L+2)-UEXT(L+1))/(UEXT(L+1)-UEXT(L))
         IF(TEST.GT.10.0) L=L+1
      END IF
C
C        *** NORMALIZE SURFACE COORDINATES SKIPPING OVER POINTS SUFFERING   ***
C        *** FROM SINGULARITIES NEAR THE CONTROL STATION (LAST THREE POINTS) **
C
      NEND=(K-2)
      IF(STAG) THEN
         S0= SC(L) - (SC(L+1)-SC(L))*UEXT(L)/(UEXT(L+1)-UEXT(L))
         SC(1)=0.0D0
         UEXT(1)=0.0D0
         NEXT=K-L
         K=1
      ELSE
         S0= 0.0D0
         NEXT=NEND
         K=0
      END IF
      XLEN=SC(NEND)-S0
      DO 110 I=L,NEND
         K=K+1
         SC(K)=(SC(I)-S0)/XLEN
         UEXT(K)=UEXT(I)
110   CONTINUE
      RETURN
      END
```

```
      FUNCTION V(S,SD,Y)
C
C**************************************************************************
C                                                                         *
C   FUNCTION V COMPUTES THE VERTICAL COMPONENT OF VELOCITY FROM THE VISCOUS *
C SOLUTION.                                                               *
C                                                                         *
C   *** LATEST REVISION - 25 JAN 1987 ***                                 *
C                                                                         *
C   *** PARAMETER DESCRIPTION ***                                         *
C                                                                         *
C     INPUT:                                                              *
C  S     - VECTOR CONTAINING THE VALUES OF THE JET PARAMETERS             *
C  SD    - VECTOR CONTAINING THE VALUES OF THE DERIVATIVES OF THE JET     *
C          PARAMETERS                                                     *
C  Y     - CARTESIAN COORDINATE (VERTICAL DISTANCE FROM JET CENTERLINE)   *
C                                                                         *
C     OUTPUT:                                                             *
C  V     - VERTICAL COMPONENT OF VELOCITY                                 *
C                                                                         *
C**************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(4),SD(4)
C
      ALP=DSQRT(DLOG(2.0D0))
      PI=3.1415926D0
C
      U0=S(1)
      U1=S(2)
      P =S(3)
      B =S(4)
C
      U0DOT=SD(1)
      U1DOT=SD(2)
      PDOT =SD(3)
      BDOT =SD(4)
C
      ETA=ALP*Y/B
      F=DSQRT(PI)/2.0D0*DERF(ETA)
      E=DEXP(-ETA**2)
C
      V=-B/ALP*(ETA*U0DOT+F*U1DOT+U1/B*(F-ETA*E)*BDOT)
C
      RETURN
      END
```

B.3.2 Dual-Jet Library TWINLIB

```
      SUBROUTINE DERIV1(M,X,S,SD)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE DERIV1 COMPUTES THE DERIVATIVES OF THE JET PARAMETERS WITHIN *
C THE VISCOUS-INVISCID MATCHING REGION.                                      *
C                                                                            *
C   *** LATEST REVISION - 23 APR 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C     INPUT:                                                                 *
C M      - NUMBER OF JET PARAMETERS                                          *
C X      - DISTANCE FROM THE JET ORIGIN                                      *
C S      - VECTOR CONTAINING THE JET PARAMETERS U0,U1,P,A,B,Y1,Y1DOT         *
C          RESPECTIVELY                                                      *
C                                                                            *
C     OUTPUT:                                                                *
C SD     - DERIVATIVES OF THE JET PARAMETERS                                 *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(7),SD(7),W(2,2),RHS(2),C(2,2)
      COMMON U0,U1,P,A,B,Y1,ALP
C
C        *** DECODE THE S ARRAY SO THE PARAMETERS MAY BE SENT IN COMMON     ***
C
      U0=S(1)
      U1=S(2)
      P=S(3)
      A=S(4)
      B=S(5)
      Y1=S(6)
      DY1DX=S(7)
C
C        *** COMPUTE THE CURVATURE OF THE JET CENTERLINE                    ***
C
      CALL FORCE1(X,DY1DX,U0DOT,ADOT,D2Y1DX)
C
C        *** CON IS A REPEATEDLY USED CONSTANT.  RK IS THE EDDY VISCOSITY   ***
C        *** SCALING CONSTANT.  RMU IS THE EDDY VISCOSITY COEFFICIENT       ***
C
      CON=1.0D0-DLOG(2.0D0)/2.0D0
      RK=0.0283D0
      RMU=RK*U1*B
C
C        *** COMPUTE THE MATRIX ELEMENTS AND RIGHT HAND SIDE
C
      D1U0  =2.0D0*U1
      C(1,1)=U0+4.0D0/3.0D0*U1+0.5D0*A
      D1A   =U1-0.5D0*CON*A
      C(1,2)=1.0D0/B*(U1*(U0+2.0D0/3.0D0*U1+0.5D0*A)-0.25D0*CON*A**2)
      T1=0.0D0
      RHS(1)=T1-(D1U0*U0DOT+D1A*ADOT)
C
      D2U0  =2.0D0*U1*(2.0D0*U0+U1+A)-0.25D0*A**2
      C(2,1)=2.0D0*U0*(U0+2.0D0*U1+A)+U1*(1.6D0*U1+2.0D0*A)+0.5D0*A**2
      D2A   =U1*(U1+2.0D0*U0+A)-CON*A*(U0+0.5D0*A)-0.125D0*A**2
      C(2,2)=1.0D0/B*(U1**2*(2.0D0*U0+8.0D0/15.0D0*U1+A)+
```

```
     1                2.0D0*U0*U1*(U0+A)+
     2                0.5D0*A**2*(-CON*(U0+0.5D0*A)+U1))
      T2=-RMU*(ALP/B)**2*(16.0D0/15.0D0*U1**2+1.0D0/3.0D0*A**2)
      RHS(2)=T2-(D2U0*U0DOT+D2A*ADOT)
C
C        *** SOLVE THE LINEAR SYSTEM FOR THE DERIVATIVES OF THE JET        ***
C        *** PARAMETERS                                                     ***
C
      CALL SIMQ(C,W,RHS,2,2,IER)
C
C        *** LOAD THE DERIVATIVES OF THE JET PARAMETERS INTO THE SD ARRAY   ***
C
      SD(1)=U0DOT
      SD(2)=RHS(1)
      SD(3)=-U0*U0DOT
      SD(4)=ADOT
      SD(5)=RHS(2)
      SD(6)=DY1DX
      SD(7)=D2Y1DX
C
      RETURN
      END
```

```
      SUBROUTINE DERIV2(M,X,S,SD)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE DERIV2 COMPUTES THE DERIVATIVES OF THE JET PARAMETERS WITHIN *
C THE FULLY VISCOUS REGION.                                                  *
C                                                                            *
C   *** LATEST REVISION - 23 APR 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C     INPUT:                                                                 *
C M      - NUMBER OF JET PARAMETERS                                          *
C X      - DISTANCE FROM THE JET ORIGIN                                      *
C S      - VECTOR CONTAINING THE JET PARAMETERS: U0,U1,P,A,B,Y1,DRAG         *
C     OUTPUT:                                                                *
C SD     - VECTOR CONTAINING THE DERIVATIVES OF THE JET PARAMETERS           *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP
      DIMENSION S(7),SD(7),W(6,6),RHS(6),C(6,6),WK(6),D(35),SUM(35)
      COMMON U0,U1,P,A,B,Y1,ALP
C
C        *** AREA 18 IS SHARED WITH PERFOR AND TWOJET.                       ***
C        *** AREA 21 IS SHARED WITH DUOAUG AND TWOJET.                       ***
C        *** ERROR IS SHARED WITH   DUOAUG AND TWOJET.                       ***
C
      COMMON /AREA18/ H,HDOT
      COMMON /AREA21/ DIFSLP,XDIFF
      COMMON /ERROR/ IERROR
C
C        *** DEFINE THE INVERSE HYPERBOLIC COSINE FUNCTION                   ***
C
      DACOSH(X)=DLOG(X+DSQRT(X**2-1.0D0))
C
C        *** ABORT IF AN ERROR CONDITION EXISTS                              ***
C
      IF(IERROR.EQ.1) GOTO 200
C
C        *** DECODE THE S VECTOR SO THE VALUES MAY BE SENT IN COMMON         ***
C
      U0=S(1)
      U1=S(2)
      P=S(3)
      A=S(4)
      B=S(5)
      Y1=S(6)
C
C        *** COMPUTE THE DIFFUSER SLOPE AND CHANNEL WIDTH                    ***
C
      IF(X.LT.XDIFF) THEN
         H=2.0D0
         HDOT=0.0D0
      ELSE
         H=2.0D0+(X-XDIFF)*DIFSLP
         HDOT=DIFSLP
      END IF
C
C        *** DEFINE REPEATEDLY USED CONSTANTS                                ***
C
```

```
      CONST=ALP/B
      C1=2.0D0*CONST*U1
      U1SQ=U1**2
      AD2=A/2.0D0
C
C        *** COMPUTE THE EDDY VISCOSITY COEFFICIENT (UCL AND UMAX ARE       ***
C        *** NORMALIZED BY U1)                                              ***
C
      ETA1=Y1*CONST
      C2=DCOSH(2.0D0*ETA1)
      C2SQ=C2**2
      UCL=4.0D0/(1.0D0+C2)
      ET=0.5D0*DACOSH(2.0D0)
      IF(ETA1.LT.ET) THEN
         ETAMAX=0.0D0
         UMAX=UCL
      ELSE
         ETAMAX=0.5D0*DACOSH((C2SQ-2.0D0)/C2)
         UMAX=C2SQ/(C2SQ-1.0D0)
      END IF
      G=4.0D0/UMAX
      F=G-1.0D0
      FSQ=F**2
      ETASTR=0.5D0*DACOSH(F*C2+DSQRT((FSQ-1.0D0)*C2SQ+2.0D0*G))
      BTILDA=(ETASTR-ETAMAX*(1.0D0-UCL/UMAX))/CONST
      UTILDA=U1*UMAX
      RK=0.0283D0
      RMU=RK*UTILDA*B
C
C        *** INITIALIZE INTEGRATION PARAMETERS.  YL IS THE LOWER LIMIT OF  ***
C        *** INTEGRATION, YU IS THE UPPER LIMIT, NINT IS THE NUMBER OF     ***
C        *** SUBINTERVALS, AND DY IS THE SUB-INTERVAL SIZE                 ***
C
      YL=0.0D0
      YU=H
      NINT=INT(4.0D0*H/B)
      IF(DMOD(DFLOAT(NINT),2.0D0).GT.0.1D0) NINT=NINT+1
      DY=(YU-YL)/DFLOAT(NINT)
C
C        *** INITIALIZE TEMPORARY STORAGE SPACE TO ZERO                    ***
C
      NVEC=35
      DO I=1,NVEC
         SUM(I)=0.0D0
      END DO
C
C        *** ENTER THE INTEGRATION LOOP ***
C
      Y=YL
      DO I=0,NINT
C
C           *** DEFINE REPEATEDLY USED TERMS                               ***
C
         ETA=CONST*Y
         ARG1=CONST*(Y+Y1)
         ARG2=CONST*(Y-Y1)
         T1=DTANH(ARG1)
         T2=DTANH(ARG2)
         S1SQ=1.0D0-T1**2
         S2SQ=1.0D0-T2**2
         T1S1SQ=T1*S1SQ
```

```
      T2S2SQ=T2*S2SQ
      G1=S1SQ*(3.0D0*S1SQ-2.0D0)
      G2=S2SQ*(3.0D0*S2SQ-2.0D0)
      Q1=DLOG(DCOSH(ARG1))
      Q2=DLOG(DCOSH(ARG2))
      SUM1=T1+T2
      SUM2=S1SQ+S2SQ
      SUM3=T1S1SQ+T2S2SQ
      SUM4=ARG1*S1SQ+ARG2*S2SQ
C
C        *** COMPUTE VELOCITY AND DERIVATIVES                            ***
C
      U=U0+AD2*(T1-T2)+U1*SUM2
      DUDY=-CONST*(AD2*(-S1SQ+S2SQ)+2.0D0*U1*SUM3)
      D2UDY2=-CONST**2*(A*(T1S1SQ-T2S2SQ)+
     1                  2.0D0*U1*(G1+G2))
C
C        *** COMPUTE THE COEFFICIENTS OF THE DERIVATIVES OF THE JET      ***
C        *** PARAMETERS                                                  ***
C
      FU0=U-ETA/CONST*DUDY
      FU1=SUM2*U-(T1+T2)/CONST*DUDY
      FP =1.0D0
      FA =0.5D0*(T1-T2)*U-0.5D0/CONST*(Q1-Q2)*DUDY
      FB =1.0D0/B*(AD2*(-ARG1*S1SQ+ARG2*S2SQ)+
     1             2.0D0*U1*(ARG1*T1S1SQ+ARG2*T2S2SQ))*U-
     2     1.0D0/ALP*(AD2*(-ARG1*T1+Q1+ARG2*T2-Q2)+
     3               U1*(-SUM4+SUM1))*DUDY
      FY1=ALP/B*(AD2*SUM2+
     1           2.0D0*U1*(-T1S1SQ+T2S2SQ))*U-
     2     (AD2*SUM1+U1*(S1SQ-S2SQ))*DUDY
      TAU=RMU*DUDY
      IF(Y.EQ.YH) TAU=0.0D0
C
C        *** ENTER A LOOP TO CYCLE THROUGH THE DIFFERENT WEIGHTING       ***
C        *** FUNCTIONS                                                   ***
C
      II=5
      DO N=0,4
         IP1=II+1
         IF(IP1.GT.5) IP1=IP1-5
         IP2=II+2
         IF(IP2.GT.5) IP2=IP2-5
         IP3=II+3
         IF(IP3.GT.5) IP3=IP3-5
         IP4=II+4
         IF(IP4.GT.5) IP4=IP4-5
         IF(N.EQ.0) THEN
            WEIGHT=FP
            DWTDY=0.0D0
         END IF
         IF(N.EQ.II) THEN
            WEIGHT=FU0/U1
            DWTDY=(-ETA/CONST*D2UDY2)/U1
         END IF
         IF(N.EQ.IP1) THEN
            WEIGHT=FU1/U1
            DWTDY=(-2.0D0*CONST*SUM3*U-
     1             1.0D0/CONST*SUM1*D2UDY2)/U1
         END IF
         IF(N.EQ.IP2) THEN
```

```
            WEIGHT=FB/U1SQ
            DWTDY=(CONST/B*(AD2*(-S1SQ*(1.0D0-2.0D0*ARG1*T1)+
     1                            S2SQ*(1.0D0-2.0D0*ARG2*T2))+
     2               2.0D0*U1*(SUM3+ARG1*G1+ARG2*G2))*U-
     3               1.0D0/ALP*(AD2*(-ARG1*T1+Q1+ARG2*T2-Q2)+
     4               U1*(-SUM4+SUM1))*D2UDY2)/U1SQ
          END IF
          IF(N.EQ.IP3) THEN
            WEIGHT=FY1/U1SQ
            DWTDY=(-CONST**2*(A*SUM3+
     1                         2.0D0*U1*(G1-G2))*U-
     3               (AD2*SUM1+U1*(S1SQ-S2SQ))*D2UDY2)/U1SQ
          END IF
          IF(N.EQ.IP4) THEN
            WEIGHT=FA/U1
            DWTDY=(0.5D0*CONST*(S1SQ-S2SQ)*U-
     1               0.5D0/CONST*(Q1-Q2)*D2UDY2)/U1
          END IF
C
C           *** LOOP TO FIND THE VALUES OF ALL OF THE INTEGRANDS         ***
C
          IND=N*7
          DO J=1,6
            IF(J.EQ.1) F=FU0
            IF(J.EQ.2) F=FU1
            IF(J.EQ.3) F=FP
            IF(J.EQ.4) F=FB
            IF(J.EQ.5) F=FY1
            IF(J.EQ.6) F=FA
            D(IND+J)=F*WEIGHT
          END DO
          D(IND+7)=-TAU*DWTDY
        END DO
C
C         *** SET THE SIMPSON'S RULE INTEGRATION WEIGHTING FACTORS       ***
C
        R=2.0D0
        IF(DMOD(DFLOAT(I),2.0D0).GT.0.1D0) R=4.0D0
        IF(I.EQ.0.OR.I.EQ.NINT) R=1.0D0
C
C         *** FIND CONTRIBUTIONS TO THE INTEGRALS                        ***
C
        DO J=1,NVEC
          SUM(J)=SUM(J)+R*D(J)
        END DO
C
C         *** INCREMENT Y                                                ***
C
        Y=Y+DY
C
      END DO
C
C       *** STORE APPROXIMATED INTEGRALS ***
C
      FACT=DY/3.0D0
      DO N=0,4
        NP1=N+1
        IND=N*7
        DO I=1,6
          C(NP1,I)=SUM(IND+I)*FACT
        END DO
```

```
         RHS(NP1)=SUM(IND+7)*FACT
      END DO
C
C        *** ENFORCE CONTINUITY ON THE UPPER WALL ***
C
      C(6,1)=-ETA/CONST
      C(6,2)=-(T1+T2)/CONST
      C(6,3)=0.0D0
      C(6,4)=-1.0D0/ALP*(AD2*(-ARG1*T1+Q1+ARG2*T2-Q2)+
     1                     U1*(-ARG1*S1SQ-ARG2*S2SQ+T1+T2))
      C(6,5)=-(AD2*(T1+T2)+U1*(S1SQ-S2SQ))
      C(6,6)=-0.5D0/CONST*(Q1-Q2)
      RHS(6)=HDOT*U
C
C        *** SOLVE THE LINEAR SYSTEM FOR THE DERIVATIVES OF THE JET PARAMETERS**
C
      D1=0.0D0
      CALL LINV3F(C,RHS,2,6,6,D1,D2,W,IER)
      IF(IER.EQ.130) THEN
         WRITE(3,107)
107      FORMAT(' ERROR IN DERIV2:  LINV3F FOUND A SINGULAR MATRIX ')
         IERROR=1
         GOTO 200
      END IF
C
      DET=D1*(2.0D0)**D2
C
C        *** LOAD THE SD VECTOR WITH THE DERIVATIVES OF THE JET PARAMETERS  ***
C
      SD(1)=RHS(1)
      SD(2)=RHS(2)
      SD(3)=RHS(3)
      SD(4)=RHS(6)
      SD(5)=RHS(4)
      SD(6)=RHS(5)
      SD(7)=SD(3)*H
C
      IERROR=0
      RETURN
C
C        *** ON ERROR CONDITION, ZERO THE JET DERIVATIVES                 ***
C
200   DO I=1,6
         SD(I)=0.0D0
      END DO
C
      RETURN
      END
```

```
      SUBROUTINE DUOBOD(XJ,YJ,DY1DX0,Y1CS,NJLS,NJLF,NJUS,NJUF,
     1                  NS,NF,IER)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE DUOBOD GENERATES THE COORDINATES OF THE EJECTOR SHROUD FOR  *
C THE DUAL JET EJECTOR.  THE SUBROUTINE READS DATA FOR THE JET TRAJECTORY    *
C CONTAINED IN LOGICAL UNIT 20 (FILE JETCL.DAT).                             *
C                                                                            *
C     *** LATEST REVISION - 23 APR 1987 ***                                  *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C     INPUT:                                                                 *
C XJ     - X COORDINATE OF THE JET NOZZLE                                    *
C YJ     - Y COORDINATE OF THE JET NOZZLE                                    *
C DY1DX0 - INITIAL JET SLOPE                                                 *
C                                                                            *
C     OUTPUT:                                                                *
C Y1CS   - Y COORDINATE OF THE JET CENTERLINE AT THE CONTROL STATION         *
C NJLS   - INDEX OF THE START OF THE JET LOWER SIDE BOUNDARY                 *
C NJLF   - INDEX OF THE FINISH OF THE JET LOWER SIDE BOUNDARY                *
C NJUS   - INDEX OF THE START OF THE JET UPPER SIDE BOUNDARY                 *
C NJUF   - INDEX OF THE FINISH OF THE JET UPPER SIDE BOUNDARY                *
C NS     - INDEX OF THE START OF THE EJECTOR SHROUD NOSE                     *
C NF     - INDEX OF THE FINISH OF THE EJECTOR SHROUD NOSE                    *
C IER    - ERROR PARAMETER 1 FOR ERROR CONDITION 0 FOR NORMAL EXECUTION      *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XCL(100),YCL(100),SPLN(300)
      DIMENSION XTMP(300)
C
C        *** DEFINE POWER-LAW STRETCHING FUNCTION                         ***
C
      COORD(I,SF,X0,X1)=((X1-X0)*SF**DFLOAT(I)-(X1-SF*X0))/(SF-1.0D0)
      RAD(DEG)=DEG/180.0D0*PI
C
      PI=3.1415926D0
      HH=0.05D0
      DX0=0.04D0
      DX1=0.3D0
      SLPJET=DTAN(RAD(9.0D0))
10    FORMAT(3F15.5)
C
C        *** READ IN THE COORDINATES OF THE JET CENTERLINE ***
C
      REWIND 20
      DO I=1,100
         READ(20,*,END=15) XCL(I),YCL(I)
      END DO
15    CONTINUE
C
C        *** EXTRAPOLATE TO GET ONE MORE POINT ***
C
      N=I-1
      DELX=5.0D0
      DYDX=(YCL(N)-YCL(N-1))/(XCL(N)-XCL(N-1))
      XCL(N+1)=XCL(N)+DELX
      YCL(N+1)=YCL(N)+DYDX*DELX
      NCL=N+1
```

```
C
C        *** SPLINE FIT THE JET CENTERLINE ***
C
      CALL ICSCCU(XCL,YCL,NCL,SPLN,NCL-1,IER)
C
C        *** DETERMINE THE COORDINATES OF THE CONTROL STATION ***
C
      X=XCL(NCL-1)+1.0D0
      TOL=1.D-4
      DO I=1,100
         CALL INTRP(X,XCL,YCL,NCL,SPLN,NCL-1,Y1,DY1DX,D2Y1DX,IER)
         IF(IER.NE.0) THEN
            WRITE(3,5) IER,X
5           FORMAT(' ERROR IN DUOBOD:  INTRP RETURNED WITH IER = ',I3,
     1             ' X = ',F10.5)
            IER=1
            RETURN
         END IF
         YB=DABS(1.0D0-Y1)+DABS(HH+(X-XJ)*SLPJET)
         RES=0.8D0-YB
         IF(I.EQ.1) THEN
            W=1.0D0
         ELSE
            W=-(X-XOLD)/(RES-RESOLD)
         END IF
         XOLD=X
         RESOLD=RES
         X=X+W*RES
         IF(DABS(RES).LT.TOL) GOTO 7
      END DO
7     CONTINUE
      XCS=X
      IF(XCS.LT.1.0D0) THEN
         WRITE(3,13)
13       FORMAT(' ERROR IN DUOBOD:  XCS WAS LESS THAN 1.0')
         IER=1
         RETURN
      END IF
      CALL INTRP(XCS,XCL,YCL,NCL,SPLN,NCL-1,Y1CS,DY1DX,D2Y1DX,IER)
      IF(IER.NE.0) THEN
         WRITE(3,5) IER,X
         IER=1
         RETURN
      END IF
      YLWR=Y1CS-(HH+(XCS-XJ)*SLPJET)
      YUPP=Y1CS+(HH+(XCS-XJ)*SLPJET)
      SLPCS=DY1DX
C
C        *** GENERATE AND TEMPORARILY STORE THE X COORDINATES ***
C        *** FOR THE JET BOUNDARY                             ***
C
      DIST=(XCS-XJ)/2.0D0
      SF=1.5D0
      X0=XJ
      X1=X0+DX0
      N=INT(DLOG(DIST/DX0*(SF-1.0D0)+1.0D0)/DLOG(SF))+1
      K=0
      IF(N.LE.5) THEN
         DO I=1,20
            F=DIST-DX0*(SF**N-1.0D0)/(SF-1.0D0)
            DF=-DX0*((DFLOAT(N)*SF**(N-1)*(SF-1.0D0)-(SF**N-1.0D0))/
```

```
     1             (SF-1.0D0)**2)
           SF=SF-F/DF
           IF(DABS(F).LT.TOL) GOTO 9
         END DO
 9       CONTINUE
         DO I=0,N
           K=K+1
           XTMP(K)=COORD(I,SF,X0,X1)
         END DO
         X0=XCS
         X1=X0-DX0
         DO I=N-1,0,-1
           K=K+1
           XTMP(K)=COORD(I,SF,X0,X1)
         END DO
         NJET=K
      ELSE
         DO I=0,5
           K=K+1
           XTMP(K)=COORD(I,SF,X0,X1)
         END DO
         X0=XCS
         X1=X0-DX0
         XP=COORD(5,SF,X0,X1)
         DIST=XP-XTMP(K)
         DX=0.3D0
         NMID=NINT(DIST/DX)
         IF(NMID.GT.0) DX=DIST/DFLOAT(NMID)
         DO I=2,NMID
           K=K+1
           XTMP(K)=XTMP(K-1)+DX
         END DO
         DO I=5,0,-1
           K=K+1
           XTMP(K)=COORD(I,SF,X0,X1)
         END DO
         NJET=K
      END IF
C
C        *** GENERATE UP TO THE LOWER CONTROL STATION ***
C
      L=0
C
      X0=0.0D0
      X1=X0-DX1
      SF=1.6D0
      DIST=20.0D0
      NPTS=NINT(DLOG((SF-1.0D0)*DIST/DX1+1.0D0)/DLOG(SF))
C
      Y=0.0D0
      VN=0.0D0
      DO I=NPTS,0,-1
         X=COORD(I,SF,X0,X1)
         L=L+1
         WRITE(1,10)X,Y,VN
      END DO
C
      SF=1.2D0
      A=DATAN(SLPJET)-DATAN(SLPCS)
      ANG=PI/2.0D0-A
      R=YLWR/(1.0D0+DSIN(ANG))
```

```
      XC=XCS-R*DCOS(ANG)
      X0=XC
      X1=X0-DX0
      NN=NINT(DLOG(DX1/DX0)/DLOG(SF))
C
      XT=COORD(NN,SF,X0,X1)
      DIST=XT-0.0D0
      IF(DIST.LT.0.0D0) THEN
         NN=NINT(DLOG((SF-1.0D0)*X0/DX0+1.0D0)/DLOG(SF))
      END IF
      N=NINT(DIST/DX1)
      IF(N.GT.0) DX=DIST/DFLOAT(N)
      DO I=1,N
         X=X+DX
         L=L+1
         WRITE(1,10)X,Y,VN
      END DO
C
      DO I=NN-1,1,-1
         X=COORD(I,SF,X0,X1)
         L=L+1
         WRITE(1,10) X,Y,VN
      END DO
C
C        *** GENERATE THE POINTS FOR THE LOWER CONTROL STATION ***
C
      A=DATAN(SLPJET)-DATAN(SLPCS)
      ANG=PI/2.0D0-A
      R=YLWR/(1.0D0+DSIN(ANG))
      XC=XCS-R*DCOS(ANG)
      YC=R
C
      DTH0=DX0/R
      THETA0=-PI/2.0D0
      THETA1=THETA0+DTH0
      DTH=PI/2.0D0-A/2.0D0
      SF=0.5D0*R+0.95D0
      N=NINT(DLOG((SF-1.0D0)*DTH/DTH0+1.0D0)/DLOG(SF))
      DO I=1,20
         F=DTH-DTH0*(SF**N-1.0D0)/(SF-1.0D0)
         DF=-DTH0*((DFLOAT(N)*SF**(N-1)*(SF-1.0D0)-(SF**N-1.0D0))/
     1           (SF-1.0D0)**2)
         SF=SF-F/DF
         IF(DABS(F).LT.TOL) GOTO 6
      END DO
C
6     CONTINUE
      DO I=0,N
         THETA=COORD(I,SF,THETA0,THETA1)
         X=XC+R*DCOS(THETA)
         Y=YC+R*DSIN(THETA)
         IF(I.EQ.N) THEN
            THNEXT=THETA+(THETA-COORD(I-1,SF,THETA0,THETA1))
         ELSE
            THNEXT=COORD(I+1,SF,THETA0,THETA1)
         END IF
         THMID=(THETA+THNEXT)/2.0D0
        VN=DCOS(THMID)
         L=L+1
         WRITE(1,10) X,Y,VN
      END DO
```

```
C
      THETA0=ANG
      THETA1=THETA0-DTH0
      DO I=N-1,1,-1
         THETA=COORD(I,SF,THETA0,THETA1)
         X=XC+R*DCOS(THETA)
         Y=YC+R*DSIN(THETA)
         THNEXT=COORD(I-1,SF,THETA0,THETA1)
         THMID=(THETA+THNEXT)/2.0D0
         VN=DCOS(THMID)
         L=L+1
         WRITE(1,10) X,Y,VN
      END DO
C
      NJLS=L+1
C
C        *** GENERATE THE POINTS ALONG THE JET BOUNDARY ***
C
      DO I=NJET,2,-1
         X=XTMP(I)
         CALL INTRP(X,XCL,YCL,NCL,SPLN,NCL-1,Y1,DY1DX,D2Y1DX,IER)
         IF(IER.NE.0) THEN
            WRITE(3,5) IER,X
            IER=1
            RETURN
         END IF
         Y=Y1-(HH+(X-XJ)*SLPJET)
         XNEXT=XTMP(I-1)
         XMID=(X+XNEXT)/2.0D0
         VN=DSIN(DATAN(SLPJET)-DATAN(DY1DX))
         L=L+1
         WRITE(1,10) X,Y,VN
      END DO
      NJLF=L+1
C
      NCIRC=4
      RR=HH/DCOS(RAD(12.0D0))
      XC=XJ+RR*DSIN(RAD(12.0D0))
      YC=YJ
      DELTH=PI-2.0D0*RAD(12.0D0)
      DTH=DELTH/DFLOAT(NCIRC)
      THETA=-PI/2.0D0-RAD(12.0D0)+DY1DX0
      VN=0.0D0
      DO I=0,NCIRC-1
         X=XC+RR*DCOS(THETA)
         Y=YC+RR*DSIN(THETA)
         L=L+1
         WRITE(1,10) X,Y,VN
         THETA=THETA-DTH
      END DO
C
      NJUS=L
      DO I=1,NJET-1
         X=XTMP(I)
         IF(I.EQ.1) THEN
            Y1=YJ
         ELSE
            CALL INTRP(X,XCL,YCL,NCL,SPLN,NCL-1,Y1,DY1DX,D2Y1DX,IER)
            IF(IER.NE.0) THEN
               WRITE(3,5) IER,X
               IER=1
```

```
                RETURN
             END IF
          END IF
          Y=Y1+(HH+(X-XJ)*SLPJET)
          XNEXT=XTMP(I+1)
          XMID=(X+XNEXT)/2.0D0
          VN=DSIN(DATAN(SLPJET)+DATAN(DY1DX))
          L=L+1
          WRITE(1,10) X,Y,VN
       END DO
       NJUF=L
C
C           *** GENERATE POINTS FOR THE UPPER CONTROL STATION ***
C
       A=DATAN(SLPJET)+DATAN(SLPCS)
       ANG=A-PI/2.0D0
       R=(2.0D0-YUPP)/(1.0D0-DSIN(ANG))
       XC=XCS-R*DCOS(ANG)
       YC=2.0D0-R
C
       DTH0=DX0/R
       THETA0=ANG
       THETA1=THETA0+DTH0
       DTH=PI/2.0D0-A/2.0D0
       SF=0.5D0*R+0.95D0
       N=NINT(DLOG((SF-1.0D0)*DTH/DTH0+1.0D0)/DLOG(SF))
       DO I=1,20
          F=DTH-DTH0*(SF**N-1.0D0)/(SF-1.0D0)
          DF=-DTH0*((DFLOAT(N)*SF**(N-1)*(SF-1.0D0)-(SF**N-1.0D0))/
     1            (SF-1.0D0)**2)
          SF=SF-F/DF
          IF(DABS(F).LT.TOL) GOTO 8
       END DO
C
8      CONTINUE
       DO I=0,N
          THETA=COORD(I,SF,THETA0,THETA1)
          X=XC+R*DCOS(THETA)
          Y=YC+R*DSIN(THETA)
          IF(I.EQ.N) THEN
             THNEXT=THETA+(THETA-COORD(I-1,SF,THETA0,THETA1))
          ELSE
             THNEXT=COORD(I+1,SF,THETA0,THETA1)
          END IF
          THMID=(THETA+THNEXT)/2.0D0
          VN=DCOS(THMID)
          L=L+1
          WRITE(1,10) X,Y,VN
       END DO
C
       THETA0=PI/2.0D0
       THETA1=THETA0-DTH0
       DO I=N-1,1,-1
          THETA=COORD(I,SF,THETA0,THETA1)
          X=XC+R*DCOS(THETA)
          Y=YC+R*DSIN(THETA)
          THNEXT=COORD(I-1,SF,THETA0,THETA1)
          THMID=(THETA+THNEXT)/2.0D0
          VN=DCOS(THMID)
          L=L+1
          WRITE(1,10) X,Y,VN
```

```
      END DO
C
C         *** GENERATE POINTS ON THE UPPER CHANNEL WALL ***
C
      X0=XC
      X1=X0-DX0
      Y=2.0D0
      VN=0.0D0
      SF=1.2D0
      DIST=X0-1.0D0
      NN=NINT(DLOG(DX1/DX0)/DLOG(SF))
      N=INT(DLOG(DIST/DX0*(SF-1.0D0)+1.0D0)/DLOG(SF))+1
      K=0
      IF(N.LE.NN) THEN
         DO I=1,20
            F=DIST-DX0*(SF**N-1.0D0)/(SF-1.0D0)
            DF=-DX0*((DFLOAT(N)*SF**(N-1)*(SF-1.0D0)-(SF**N-1.0D0))/
     1                (SF-1.0D0)**2)
            IF(DABS(DF).LT.1.D-6) THEN
               SF=SF-F
            ELSE
               SF=SF-F/DF
            END IF
            IF(DABS(F).LT.TOL) GOTO 11
         END DO
11       CONTINUE
         DO I=0,N-1
            X=COORD(I,SF,X0,X1)
            L=L+1
            WRITE(1,10) X,Y,VN
         END DO
      ELSE
         DO I=0,NN
           X=COORD(I,SF,X0,X1)
           L=L+1
           WRITE(1,10) X,Y,VN
         END DO
         DIST=X-1.0D0
         N=NINT(DIST/DX1)
         IF(N.NE.0) DX=DIST/DFLOAT(N)
         DO I=1,N-1
            X=X-DX
            L=L+1
            WRITE(1,10) X,Y,VN
         END DO
      END IF
C
C         *** GENERATE THE POINTS FOR THE BODY NOSE ***
C
      NCIRC=12
      XC=1.0D0
      YC=3.0D0
      R=1.0D0
      DTH=PI/DFLOAT(NCIRC)
      THETA=3.0D0/2.0D0*PI
      VN=0.0D0
      NS=L+1
      DO 110 I=0,NCIRC
         X=XC+R*DCOS(THETA)
         Y=YC+R*DSIN(THETA)
         L=L+1
```

```
            WRITE(1,10) X,Y,VN
            THETA=THETA-DTH
110     CONTINUE
C
        NF=L-1
C
        DX=0.2D0
        SF=1.2D0
        X0=X
        X1=X0+DX
        DIST=XCS-X
        N=NINT(DLOG(DIST/DX*(SF-1.0D0)+1.0D0)/DLOG(SF))
        DO I=1,N
            X=COORD(I,SF,X0,X1)
            L=L+1
            WRITE(1,10) X,Y,VN
        END DO
C
        DX=COORD(N,SF,X0,X1)-COORD(N-1,SF,X0,X1)
        X0=X-DX
        X1=X
        SF=1.6D0
        DIST=20.0D0-X0
        N=NINT(DLOG(DIST/DX*(SF-1.0D0)+1.0D0)/DLOG(SF))
        DO I=2,N
            X=COORD(I,SF,X0,X1)
            L=L+1
            WRITE(1,10) X,Y,VN
        END DO
C
        IER=0
C
        RETURN
        END
```

```
      SUBROUTINE FORCE1(X,DY1DX,UODOT,ADOT,D2Y1DX)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE FORCE1 COMPUTES THE CURVATURE OF THE JET CENTERLINE.        *
C                                                                            *
C   *** LATEST REVISION - 23 APR 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C     INPUT:                                                                 *
C X      - DISTANCE FROM THE JET ORIGIN                                      *
C DY1DX  - JET CENTERLINE SLOPE                                              *
C                                                                            *
C     OUTPUT:                                                                *
C UODOT  - DERIVATIVE OF THE EXTERNAL VELOCITY                               *
C ADOT   - DERIVATIVE OF THE ASYMMETRY FACTOR                                *
C D2Y1DX - SECOND DERIVATIVE OF THE JET CENTERLINE                           *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON U0,U1,P,A,B,Y1,ALP
      EXTERNAL USQ
C
C       *** INITIALIZE INTEGRATION PARAMETERS                              ***
C
      ETAMAX=2.4D0
      YL=Y1-ETAMAX*B/ALP
      YU=Y1+ETAMAX*B/ALP
      NINT=20
C
C        *** COMPUTE PRIMARY JET MOMENTUM                                  ****
C
      RMJ=SIMS(USQ,YL,YU,NINT)
C
C        *** FIND THE VELOCITY COMPONENTS ON EITHER SIDE OF THE JET        ***
C
      CALL UPPVLC(X,UU,UUDOT)
      CALL LWRVLC(X,UL,ULDOT)
C
C        *** COMPUTE THE PRESSURE JUMP ACROSS THE JET                      ***
C
      DELP=0.5D0*(UL*UL-UU*UU)
C
C        *** COMPUTE THE CURVATURE OF THE JET CENTERLINE                   ***
C
      RKAP=-DELP/RMJ
C
      UODOT=UUDOT
      ADOT=ULDOT-UUDOT
C
C        *** COMPUTE THE SECOND DERIVATIVE OF THE JET CENTERLINE           ***
C
      D2Y1DX=RKAP*(1+DY1DX**2)**1.5D0
C
      RETURN
      END
```

```
      SUBROUTINE FSTRM(U10,BO,UOO,AO,GAMMA,VO)
C
C***************************************************************************
C                                                                          *
C     SUBROUTINE FSTRM COMPUTES THE VALUE OF THE FREE STREAM VELOCITY GIVEN *
C THE PARAMETER GAMMA AND THE VALUES OF THE JET PARAMETERS.                *
C                                                                          *
C   *** LATEST REVISION - 23 APR 1987 ***                                  *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C     INPUT:                                                               *
C U10     - INITIAL JET EXCESS VELOCITY                                    *
C BO      - INITIAL JET HALF-WIDTH                                         *
C AO      - INITIAL ASYMMETRY FACTOR                                       *
C GAMMA   - FREE STREAM SPEED PARAMETER                                    *
C                                                                          *
C     OUTPUT:                                                              *
C VO      - FREE STREAM VELOCITY                                           *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z  )
      COMMON UO,U1,P,A,B,Y1,ALP
C
C        *** COMPUTE THE PRIMARY JET MOMENTUM FLUX                      ***
C
      RMJ=2.ODO*BO/ALP*(2.ODO/3.ODO*U10**2+2.ODO*UOO*U10+U10*AO-
     1                  AO**2/4.ODO)
C
C        *** COMPUTE THE FREE STREAM VELOCITY                           ***
C
      VO=GAMMA*DSQRT(RMJ/4.ODO)
C
      RETURN
      END
```

```
      SUBROUTINE GETPRM(XJ,YJ,Y1DOTO,XEXIT,DIFSLP,XDIFF,GAMMA,
     1                 U1O,BO,DUMP1)
C
C*****************************************************************************
C                                                                            *
C     THIS SUBROUTINE READS INPUTS FROM DATA FILE CASE.DAT.   THE INFORMATION *
C ACQUIRED PERTAINS TO THE DETAILS OF THE SHROUD BODY AS WELL AS THE FLOW    *
C CONDITIONS.                                                                *
C                                                                            *
C   *** LATEST REVISION - 1 FEB 1987 ***                                     *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     OUTPUT:                                                                *
C  XJ    - X COORDINATE OF THE JET NOZZLE POSITION                           *
C  YJ    - Y COORDINATE OF THE JET NOZZLE                                    *
C  Y1DOTO INITIAL SLOPE OF THE JET CENTERLINE                                *
C  XEXIT- X COORDINATE OF THE SHROUD EXIT                                    *
C  DIFSLP DIFFUSER SLOPE                                                     *
C  XDIFF- X COORDINATE OF THE START OF THE DIFFUSER                          *
C  GAMMA- FREE-STREAM SPEED PARAMETER                                        *
C  U1O  - JET INITIAL CENTERLINE VELOCITY                                    *
C  BO   - INITIAL JET HALF-WIDTH                                             *
C  DUMP1- LOGICAL PARAMETER TO CONTROL OUTPUT                                *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP1
C
      READ(4,*) XJ
      READ(4,*) YJ
      READ(4,*) Y1DOTO
      READ(4,*) XEXIT
      READ(4,*) DIFSLP
      READ(4,*) XDIFF
      READ(4,*) GAMMA
      READ(4,*) U1O
      READ(4,*) BO
      READ(4,*) DUMP1
C
      RETURN
      END
```

```
      SUBROUTINE JETMAT(NJLS,NJLF,NJUS,NJUF,XCP,YCP,ALPHA,D,IND1,IND2,
     1                  PD,PE,PF,PG,PH,PPI,C,WORK,A,B,AMAT,BMAT,N,
     2                  ALWR,BLWR,AUPP,BUPP)
C
C*************************************************************************
C                                                                        *
C   SUBROUTINE JETMAT COMPUTES THE INFLUENCE COEFFICIENTS FOR THE MATCHING   *
C POINTS ALONG THE JET BOUNDARIES.                                       *
C                                                                        *
C   *** LATEST REVISION - 23 APR 1987 ***                                *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  NJLS  - PANEL NUMBER OF THE BEGINNING OF THE LOWER JET BOUNDARY       *
C  NJLF  - PANEL NUMBER OF THE END OF THE LOWER JET BOUNDARY             *
C  NJUS  - PANEL NUMBER OF THE BEGINNING OF THE UPPER JET BOUNDARY       *
C  NJUF  - PANEL NUMBER OF THE END OF THE UPPER JET BOUNDARY             *
C  XCP   - VECTOR OF CONTROL POINT X COORDINATES                         *
C  YCP   - VECTOR OF CONTROL POINT Y COORDINATES                         *
C  ALPHA - VECTOR CONTAINING THE SURFACE SLOPES                          *
C  D     - VECTOR CONTAINING THE PANEL LENGTHS                           *
C  IND1  - VECTOR OF INDEX OF PANEL ADJOINING TO THE LEFT                *
C  IND2  - VECTOR OF INDEX OF PANEL ADJOINING TO THE RIGHT               *
C PD..PPI- SOURCE PARABOLIC FIT COEFFICIENTS                             *
C  C     - VECTOR OF SURFACE CURVATURE COEFFICIENTS                      *
C  WORK  - WORK SPACE VECTOR                                             *
C  A     - WORK SPACE VECTOR                                             *
C  B     - WORK SPACE VECTOR                                             *
C  AMAT  - MATRIX OF X COMPONENT INDUCED VELOCITIES                      *
C  BMAT  - MATRIX OF Y COMPONENT INDUCED VELOCITIES                      *
C  N     - NUMBER OF PANELS                                              *
C                                                                        *
C     OUTPUT:                                                            *
C  ALWR  - MATRIX OF U-VELOCITY INFLUENCE COEFFICIENTS FOR THE LOWER BOUNDARY *
C  BLWR  - MATRIX OF V-VELOCITY INFLUENCE COEFFICIENTS FOR THE LOWER BOUNDARY *
C  AUPP  - MATRIX OF U-VELOCITY INFLUENCE COEFFICIENTS FOR THE UPPER BOUNDARY *
C  BUPP  - MATRIX OF V-VELOCITY INFLUENCE COEFFICIENTS FOR THE UPPER BOUNDARY *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XCP(N),YCP(N),ALPHA(N),D(N),PD(N),PE(N),PF(N),PG(N),
     1          PH(N),PPI(N),C(N),IND1(N),IND2(N),WORK(8*N),
     2          A(N),B(N),AMAT(N,N),BMAT(N,N),
     3          ALWR(NJLF-NJLS+1,N),BLWR(NJLF-NJLS+1,N),
     4          AUPP(NJUF-NJUS+1,N),BUPP(NJUF-NJUS+1,N)
C
C        *** AREA15 IS SHARED WITH LWRVLC, ONEJET, AND VLCJET ***
C        *** AREA16 IS SHARED WITH UPPVLC, ONEJET, AND VLCJET ***
C
      COMMON /AREA15/ XL(100),YL(100),UL(100),VL(100),SPLNUL(100,3),NL
      COMMON /AREA16/ XU(100),YU(100),UU(100),VU(100),SPLNUU(100,3),NU
C
C        *** CALCULATE AND STORE INFLUENCE COEFFICIENTS FOR THE LOWER ***
C        *** JET BOUNDARY                                             ***
C
      NL=NJLF-NJLS+1
C
      DO I=NJLF,NJLS,-1
C
```

```
         II=NJLF-I+1
         XL(II)=XCP(I)
         YL(II)=YCP(I)
C
         DO J=1,N
            ALWR(II,J)=AMAT(I,J)
            BLWR(II,J)=BMAT(I,J)
         END DO
C
C           *** USE A POINT SLIGHTLY OFF THE JET BOUNDARY WHEN COMPUTING    ***
C           *** THE VELOCITIES NEAR THE CONTROL STATION TO AVOID THE SPIKE  ***
C           *** CAUSED BY THE CURVATURE DISCONTINUITY                       ***
C
         IF(I.LT.(NJLS+5)) THEN
            YM=YCP(I)-(XCP(I)-XCP(NJLS+5))*0.15D0
            CALL INFLCE(XCP(I),YM,XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,
     1                     PF,PG,PH,PPI,C,WORK,N,A,B)
            DO J=1,N
               ALWR(II,J)=A(J)
            END DO
C
         END IF
C
      END DO
C
C        *** CALCULATE AND STORE THE INFLUENCE COEFFICIENTS FOR THE ***
C        *** UPPER BOUNDARY                                         ***
C
      NU=NJUF-NJUS+1
      DO I=NJUS,NJUF
C
         II=I-NJUS+1
         XU(II)=XCP(I)
         YU(II)=YCP(I)
C
         DO J=1,N
            AUPP(II,J)=AMAT(I,J)
            BUPP(II,J)=BMAT(I,J)
         END DO
C
C           *** USE A POINT SLIGHTLY OFF THE JET BOUNDARY WHEN COMPUTING    ***
C           *** THE VELOCITIES NEAR THE CONTROL STATION TO AVOID THE SPIKE  ***
C           *** CAUSED BY THE CURVATURE DISCONTINUITY                       ***
C
         IF(I.GT.(NJUF-5)) THEN
            YM=YCP(I)+(XCP(I)-XCP(NJUF-5))*0.15D0
            CALL INFLCE(XCP(I),YM,XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,
     1                  PF,PG,PH,PPI,C,WORK,N,A,B)
            DO J=1,N
               AUPP(II,J)=A(J)
            END DO
C
         END IF
C
      END DO
C
C        *** STANDARDIZE THE X COORDINATE VECTORS ***
C
      DO I=1,NL
         XU(I)=XL(I)
      END DO
```

```
C
      RETURN
      END
```

```
      SUBROUTINE LWRVLC(X,U,UDOT)
C
C******************************************************************************
C                                                                             *
C     SUBROUTINE LWRVLC COMPUTES THE HORIZONTAL COMPONENT OF VELOCITY ON THE  *
C LOWER SIDE OF THE JET BOUNDARY.                                             *
C                                                                             *
C   *** LATEST REVISION - 24 APR 1987 ***                                     *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C     INPUT:                                                                  *
C X     - DISTANCE FROM THE JET ORIGIN                                        *
C     OUTPUT                                                                  *
C U     - HORIZONTAL COMPONENT OF VELOCITY AT THE STATION X                   *
C UDOT  - dU/dx AT THE STATION X                                              *
C                                                                             *
C******************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C        *** AREA15 IS SHARED WITH JETMAT, JETVLC AND ONEJET ***
C
      COMMON /AREA15/ XL(100),YL(100),UL(100),VL(100),SPLNUL(100,3),NL
C
      CALL INTRP(X,XL,UL,NL,SPLNUL,100,U,UDOT,D2UDX2,IER)
      IF(IER.NE.0) THEN
         WRITE(3,10) IER,X
10       FORMAT(' ERROR IN LWRVLC:  INTRP RETURNED WITH IER = ',I3,
     1          ' X = ',F10.5)
         STOP
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE ONEJET(NJLS,NJLF,NJUS,NJUF,YJ,Y1DOTO,U1O,BO,VO,
     1                  ALPHA,VN,N,DUMP,NCALL,UOO,AO,XEND,Y1END,RES)
C
C***************************************************************************
C                                                                          *
C     SUBROUTINE ONEJET PERFORMS THE VISCOUS CALCULATION WITHIN THE VISCOUS- *
C INVISCID INTERACTION REGION.  THE DERIVATIVE OF UO IS FOUND FROM THE     *
C INVISCID SOLUTION VIA A SPLINE FIT, AND IS USED AS A FORCING TERM IN     *
C THE VISCOUS SOLUTION.                                                    *
C                                                                          *
C   *** LATEST REVISION - 24 APR 1987 ***                                  *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C     INPUT:                                                               *
C  NJLS  PANEL NUMBER OF THE BEGINNING OF THE LOWER JET BOUNDARY           *
C  NJLF  PANEL NUMBER OF THE END OF THE LOWER JET BOUNDARY                 *
C  NJUS  PANEL NUMBER OF THE BEGINNING OF THE UPPER JET BOUNDARY           *
C  NJUF  PANEL NUMBER OF THE END OF THE UPPER JET BOUNDARY                 *
C  YJ - JET INITIAL Y COORDINATE                                           *
C  Y1DOTO JET CENTERLINE INITIAL SLOPE                                     *
C  U1O - JET INITIAL CENTERLINE VELOCITY                                   *
C  BO  - JET INITIAL HALF-WIDTH                                            *
C  VO  - FREE STREAM VELOCITY                                              *
C  ALPHA PANEL ORIENTATION ANGLES                                          *
C  VN  - VECTOR CONTAINING THE NORMAL VELOCITIES TO THE PANELS ALONG THE JET *
C        BOUNDARY IN THE VISCOUS-INVISCID INTERACTION REGION               *
C  N   - NUMBER OF PANELS                                                  *
C  DUMP- LOGICAL PARAMETER FOR CONTROLLING OUTPUT                          *
C  NCALL INDEX TO KEEP TRACK OF SUCCESSIVE CALLS TO ONEJET                 *
C                                                                          *
C     OUTPUT:                                                              *
C  UOO - VALUE OF UO AT THE JET EXIT                                       *
C  AO  - VALUE OF A AT THE JET EXIT                                        *
C  VN -  UPDATED NORMAL VELOCITY VECTOR                                    *
C  XEND- X STATION AT WHICH THE VISCOUS-INVISCID MATCHING ENDS             *
C  Y1END VALUE OF Y1 AT XEND                                               *
C  RES - MAXIMUM RESIDUAL IN THE VISCOUS-INVISCID MATCHING                 *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP
      DIMENSION S(7),SD(7),RD(6),W(7,9),C(24),ALPHA(N),VN(N)
      COMMON UO,U1,P,A,B,Y1,ALP
C
C        *** AREA15 IS SHARED WITH JETMAT, LWRVLC, AND VLCJET ***
C        *** AREA16 IS SHARED WITH JETMAT, UPPVLC, AND VLCJET ***
C
      COMMON /AREA15/ XL(100),YL(100),UL(100),VL(100),SPLNUL(100,3),NL
      COMMON /AREA16/ XU(100),YU(100),UU(100),VU(100),SPLNUU(100,3),NU
C
      EXTERNAL DERIV1
C
      IF(DUMP) THEN
         REWIND 9
         REWIND 10
         REWIND 11
      END IF
C
      REWIND 20
```

```
C
      N=7
      NW=7
      TOL=.001D0
      IND=1
C
C        *** OBTAIN THE INTERPOLATED VALUE OF THE HORIZONTAL COMPONENT OF ***
C        *** INVISCID VELOCITY AT THE JET NOZZLE                         ***
C
      X=XU(2)-.001
      CALL UPPVLC(X,UUU,UUDOT)
      CALL LWRVLC(X,ULL,ULDOT)
      UOO=UUU
      AO=ULL-UUU
C
C        *** DEFINE INITIAL VALUES OF THE JET PARAMETERS ***
C
      S(1)=UOO
      S(2)=U10
      S(3)=0.0D0
      S(4)=AO
      S(5)=BO
      S(6)=YJ
      S(7)=Y1DOTO
      RES=0.0
C
C        *** ENTER LOOP TO MARCH THE VISCOUS EQUATIONS
C
      NE=NJUF-NJUS+1
C
      IF(DUMP) THEN
         WRITE(9,35)
35       FORMAT(/,25X,' LOWER JET VELOCITIES ')
         WRITE(9,40)
40       FORMAT(/,'       X          Y        UINV       VINV       VVIS',
     &           '      VNOLD      VNNEW       RES')
         WRITE(10,45)
45       FORMAT(/,25X, ' UPPER JET VELOCITIES ')
         WRITE(10,40)
         WRITE(11,50)
50       FORMAT(/,25X,' JET SOLUTION ')
         WRITE(11,55)
55       FORMAT(/,'       X    UO,UODOT  U1,U1DOT   P,PDOT',
     1          '   A,ADOT    B,BDOT   Y1,Y1DOT DY1D,D2Y1')
      END IF
C
      DO 10 J=2,NE
         XEND =XU(J)
         CALL DVERK(N,DERIV1,X,S,XEND,TOL,IND,C,NW,W,IER)
         IF(IND.LT.0.OR.IER.GT.0) THEN
            WRITE(3,150) IND,IER
150         FORMAT(/,' ERROR IN TWOJET, DVERK RETURNED WITH IND = ',I5,
     &             ' IER = ',I5)
            STOP
         END IF
C
C           *** OBTAIN THE LOCAL DERIVATIVES OF THE JET PARAMETERS ***
C
   CALL DERIV1(N,XEND,S,SD)
C
         WRITE(20,62) XEND,Y1
```

```
62        FORMAT(2F10.5)
          IF(DUMP) THEN
             WRITE(11,60) XU(J),(S(II),II=1,7),XL(J),(SD(II),II=1,7)
60           FORMAT(8F10.5,/8F10.5,/)
          END IF
C
C            *** COMPUTE THE VERTICAL COMPONENT OF VELOCITY AT THE JET ***
C            *** BOUNDARY FROM THE VISCOUS SOLUTION                    ***
C
          DO I=1,5
             RD(I)=SD(I)
          END DO
          RD(6)=0.0D0
C
          VVISU=V(RD,YU(J))
          VVISL=V(RD,YL(J))
C
C            *** UPDATE THE SUCTION VELOCITY ON THE LOWER JET BOUNDARY ***
C
          RL=VVISL-VL(J)
          W1=1.0D0-0.8D0/DFLOAT(NE-1)*DFLOAT(J-1)
          VNEWL= VN(NJLF-(J-1))+W1*RL
          RESL=DABS(RL)
          IF(RESL.GT.RES) RES=RESL
          IF(DUMP) THEN
             WRITE(9,30) XL(J),YL(J),UL(J),VL(J),VVISL,VN(NJLF-(J-1)),
     &                   VNEWL,RL
30           FORMAT(8F10.5)
          END IF
C
C            *** MAKE A CORRECTION TO THE LOCAL ENTRAINMENT VELOCITY ***
C
          VN(NJLF-(J-1))=VNEWL
C
C            *** UPDATE THE SUCTION VELOCITY ON THE UPPER JET BOUNDARY ***
C
          RU=VVISU-VU(J)
          W1=-(1.0D0-0.8D0/DFLOAT(NE-1)*DFLOAT(J-1))
          VNEWU= VN(NJUS+(J-1))+W1*RU
          RESU=DABS(RU)
          IF(RESU.GT.RES) RES=RESU
          IF(DUMP) THEN
             WRITE(10,30) XU(J),YU(J),UU(J),VU(J),VVISU,VN(NJUS+(J-1)),
     &                   VNEWU,RU
          END IF
C
C            *** MAKE A CORRECTION TO THE LOCAL ENTRAINMENT VELOCITY ***
C
          VN(NJUS+(J-1))=VNEWU
C
10     CONTINUE
C
       Y1END=Y1
C
       RETURN
       END
```

```
      SUBROUTINE PERFOR(ALPHA,D,AMAT,BMAT,Q,N,VO,BETA,
     1                  U10,UOO,AO,BO,DFDRAG,NS,NF,NJLF,NJUS,PHI)
C
C******************************************************************************
C                                                                             *
C     SUBROUTINE PERFOR COMPUTES THE THRUST AUGMENTATION RATIO IN TWO         *
C INDEPENDENT CALCULATIONS; THROUGH INTEGRATION OF THE SURFACE PRESSURES AND  *
C THROUGH USE OF THE BLASIUS INTEGRAL LAW.  IN THE SURFACE PRESSURE BASED     *
C CALCULATION, THE SUCTION ACTING ON BOTH THE EJECTOR SHROUD LEADING EDGE AND *
C THE PRIMARY JET NOZZLE ARE TAKEN INTO ACCOUNT.                              *
C                                                                             *
C   *** LATEST REVISION - 2 FEB 1987 ***                                      *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C                                                                             *
C     INPUT:                                                                  *
C  ALPHA - VECTOR CONTAINING THE SURFACE SLOPES                               *
C  D     - VECTOR CONTAINING THE PANEL LENGTHS                                *
C  AMAT  - MATRIX OF HORIZONTAL INDUCED VELOCITY COEFFICIENTS                 *
C  BMAT  - MATRIX OF VERTICAL INDUCED VELOCITY COEFFICIENTS                   *
C  Q     - VECTOR CONTAINING THE SOURCE STRENGTHS                             *
C  N     - NUMBER OF PANELS                                                   *
C  VO    - FREE STREAM SPEED                                                  *
C  BETA  - ANGLE OF ATTACK                                                    *
C  U10   - INITIAL JET VELOCITY                                               *
C  UOO   - INITIAL JET EXTERNAL VELOCITY                                      *
C  AO    - INITIAL ASYMMETRY FACTOR                                           *
C  BO    - INITIAL JET HALF-WIDTH                                             *
C  DFDRAG- DIFFUSER ASSOCIATED DRAG                                           *
C  NS    - INDEX OF BEGINNING OF SHROUD NOSE                                  *
C  NF    - INDEX OF END OF SHROUD NOSE                                        *
C  NJLF  - INDEX OF FINISH OF JET LOWER BOUNDARY                              *
C  NJUS  - INDEX OF START OF JET UPPER BOUNDARY                               *
C                                                                             *
C     OUTPUT:                                                                 *
C  PHI   - THRUST AUGMENTATION COMPUTED USING THE MOMENTUM THEOREM            *
C                                                                             *
C******************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALPHA(N),D(N),AMAT(N,N),BMAT(N,N),Q(N)
      COMMON U0,U1,P,A,B,Y1,ALP
C
C        *** AREA 18 IS SHARED WITH TWOJET                                 ***
C
      COMMON /AREA18/ H,HDOT
      EXTERNAL U,USQ
C
      PI=3.1415926DO
C
      UOX=VO*DCOS(BETA)
      UOY=VO*DSIN(BETA)
C
C        *** COMPUTE THE PRIMARY JET MOMENTUM                              ***
C
      RMJ=2.ODO*BO/ALP*(2.ODO/3.ODO*U10**2+2.ODO*UOO*U10+U10*AO-
     1                  AO**2/4.ODO-U10*VO)
C
C
C        *** INTEGRATE THE SURFACE PRESSURES ON THE EJECTOR SHROUD         ***
C
```

```
      SUM1=0.0D0
      SUM2=0.0D0
      K=0
      NN=NF-NS+1
      R=1.0D0
      DTH=-PI/DFLOAT(NN)
      TH=3.0D0/2.0D0*PI+DTH/2.0D0
      DO I=NS,NF
         K=K+1
         WT=2.0D0
         IF(DMOD(DFLOAT(K),2.0D0).LT.0.1D0) WT=4.0D0
         IF(K.EQ.1.OR.K.EQ.NN) WT=1.0D0
C
         SUM3=0.0D0
         SUM4=0.0D0
         DO J=1,N
            SUM3=SUM3+AMAT(I,J)*Q(J)
            SUM4=SUM4+BMAT(I,J)*Q(J)
         END DO
C
         UU=SUM3+U0X
         VV=SUM4+U0Y
         PR=0.5D0*(UU**2+VV**2-V0**2)
C
         SUM1=SUM1+PR*WT/3.0D0*DCOS(TH)*R*DTH
         SUM2=SUM2+PR*D(I)*DSIN(ALPHA(I))
         TH=TH+DTH
      END DO
C
      TS1=SUM1
      TS2=SUM2
      WRITE(21,5) TS1,TS2
5     FORMAT(//,' SHROUD THRUST SIMPSONS RULE, MIDPOINT RULE: ',2F10.5)
C
C        *** INTEGRATE THE SURFACE PRESSURE ON THE JET NOZZLE              ***
C
      SUM1=0.0D0
      SUM2=0.0D0
      NN=NJUS-NJLF+1
      R=.05D0
      DTH=-PI/DFLOAT(NN)
      TH=3.0D0/2.0D0*PI+DTH/2.0D0
      K=0
      DO I=NJLF,NJUS
         K=K+1
         WT=2.0D0
         IF(DMOD(DFLOAT(K),2.0D0).LT.0.1D0) WT=4.0D0
         IF(K.EQ.1.OR.K.EQ.NN) WT=1.0D0
C
         SUM3=0.0D0
         SUM4=0.0D0
         DO J=1,N
            SUM3=SUM3+AMAT(I,J)*Q(J)
            SUM4=SUM4+BMAT(I,J)*Q(J)
         END DO
C
         UU=SUM3+U0X
         VV=SUM4+U0Y
         PR=0.5D0*(UU**2+VV**2-V0**2)
C
         SUM1=SUM1+PR*WT/3.0D0*COS(TH)*R*DTH
```

```
            SUM2=SUM2+PR*D(I)*DSIN(ALPHA(I))
            TH=TH+DTH
         END DO
C
         TC1=SUM1
         TC2=SUM2
C
         WRITE(21,6) TC1,TC2
6        FORMAT(' NOZZLE CAP THRUST SIMPSONS RULE, MIDPOINT RULE: ',2F10.5)
C
         TAUX=TS1+TC1
C
C           *** COMPUTE THE MOMENTUM FLUX AT THE EJECTOR EXIT                   ***
C
         ETAH=3.0D0
         TP=B0/ALP*((U00**2+U00*A0+A0**2/2.0D0-V0**2)*ETAH-
        1           2.0D0*(U0*ETAH+DLOG(2.0D0)/2.0D0*A)*V0)
         PHISP=1.0D0+(TAUX+TP-DFDRAG)/RMJ
C
         TGROSS=SIMS(USQ,0.0D0,H,30)-V0*SIMS(U,0.0D0,H,30)
         PHIMT=TGROSS/RMJ
C
         TIND=TGROSS-(RMJ+TP)+DFDRAG
C
         WRITE(21,10) RMJ,TGROSS,TAUX,TIND,DFDRAG,PHISP,PHIMT
10       FORMAT(//,' JET MOMENTUM = ',F10.5,
        1          ' EXITING MOMENTUM = ',F10.5,/,
        2 ' INDUCED THRUST COMPUTED FROM SURFACE PRESSURES = ',F10.5,/,
        3 ' INDUCED THRUST COMPUTED FROM MOMENTUM THEOREM  = ',F10.5,/,
        4 ' PRESSURE DRAG ASSOCIATED WITH THE DIFFUSER     = ',F10.5,/,
        4 ' THRUST AUGMENTATION RATIO FROM SURFACE PRESSURES = ',F10.5,/,
        5 ' THRUST AUGMENTATION RATIO FROM MOMENTUM THEOREM  = ',F10.5)
C
         PHI=PHIMT
C
         RETURN
         END
```

```
      SUBROUTINE TWOJET(XEXIT,XBEGIN,DUMP,PEXIT,DFDRAG,IER)
C
C*****************************************************************************
C                                                                            *
C   SUBROUTINE TWOJET MARCHES THE VISCOUS SOLUTION WITHIN THE CHANNEL BEYOND *
C THE MATCHING REGION.                                                       *
C                                                                            *
C   *** LATEST REVISION - 24 APR 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  XEXIT  - X COORDINATE OF THE SHROUD EXIT                                  *
C  XBEGIN - X COORDINATE TO START THE MARCHING                               *
C  DUMP   - LOGICAL PARAMETER USED TO CONTROL OUTPUT                         *
C                                                                            *
C     OUTPUT:                                                                *
C  PEXIT  -  STATIC PRESSURE COMPUTED BY THE VISCOUS SOLUTION AT THE EXIT    *
C  DFDRAG -  DRAG ASSOCIATED WITH THE DIFFUSER                               *
C  IER    -  ERROR PARAMETER: 0 FOR NORMAL EXECUTION, 1 FOR ERROR            *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL DUMP
      DIMENSION S(7),SD(7),W(7,9),C(24)
      COMMON U0,U1,P,A,B,Y1,ALP
C
C         *** ERROR IS SHARED WITH DERIV2                                  ***
C         *** AREA 18 IS SHARED WITH PERFOR                                ***
C         *** AREA 21 IS SHARED WITH DUOAUG AND DERIV2                     ***
C
      COMMON /ERROR/ IERROR
      COMMON /AREA18/ H,HDOT
      COMMON /AREA21/ DIFSLP,XDIFF
C
      EXTERNAL DERIV2,USQ,U
C
      IF(DUMP) REWIND 12
C
      IERROR=0
C
C         *** INITIALIZE PARAMETERS FOR THE DVERK ROUTINE                  ***
C
      N=7
      NW=7
      TOL=1.D-4
      IND=1
C
C         *** DECODE THE S VECTOR SO THAT THE VALUES MAY BE SENT IN COMMON ***
C
      S(1)=U0
      S(2)=U1
      S(3)=P
      S(4)=A
      S(5)=B
      S(6)=Y1
      S(7)=0.0D0
C
C         *** COMPUTE THE STARTING MOMENTUM AND MASS FLUX                  ***
C
```

```
      RMJ1=SIMS(USQ,0.0D0,2.0D0,100)+2.0D0*P
      RMDOT1=SIMS(U,0.0D0,2.0D0,100)
C
      IF(DUMP) THEN
         WRITE(12,50) RMJ1,RMDOT1
50       FORMAT(/,25X,' JET IN CHANNEL SOLUTION ',/,
     1          ' INITIAL JET MOMENTUM = ',F10.5,' INITIAL MASS = ',
     2          F10.5)
         WRITE(12,55)
55       FORMAT(/,'       X      U0,U0DOT   U1,U1DOT    P,PDOT',
     1          '     A,ADOT     B,BDOT    Y1,Y1DOT ')
      END IF
C
C        *** INITIALIZE PARAMETERS FOR THE INTEGRATION OF THE VISCOUS EQS.  ***
C
      DX=0.5D0
      DIST=XEXIT-XBEGIN
      NPTS=NINT(DIST/DX)
      DX=DIST/DFLOAT(NPTS)
      X=XBEGIN
      PSTART=P
      HSTART=2.0D0
C
C        *** ENTER LOOP TO MARCH THE VISCOUS EQUATIONS                      ***
C
      DO I=1,NPTS
         XEND=X+DX
         CALL DVERK(M,DERIV2,X,S,XEND,TOL,IND,C,MW,W,IER)
         IF(IERROR.EQ.1) THEN
            IER=1
            RETURN
         END IF
         IF(IND.LT.0.OR.IER.GT.0) THEN
            WRITE(3,150) IND,IER
150         FORMAT(/,' ERROR IN TWOJET, DVERK RETURNED WITH IND = ',I5,
     &             ' IER = ',I5)
            IER=1
            RETURN
         END IF
C
C
         IF(DUMP) THEN
            CALL DERIV2(M,X,S,SD)
            WRITE(12,60) X,(S(J),J=1,6),X,(SD(J),J=1,6)
60          FORMAT(7F11.5,/,7F11.5,/)
         END IF
C
      END DO
C
C        *** STORE THE EXIT PRESSURE AND COMPUTE THE DIFFUSER PRESSURE DRAG ***
C
      PEXIT=P
      DFDRAG=S(7)-(P-PSTART)*HSTART
C
C        *** COMPUTE THE FINAL MOMENTUM AND MASS FLUX                       ***
C
      RMJ=SIMS(USQ,0.0D0,H,100)+P*HSTART+DFDRAG
      RMDOT=SIMS(U,0.0D0,H,100)
      IF(DUMP) THEN
         WRITE(12,70) RMJ,RMDOT
70       FORMAT(' FINAL MOMENTUM = ',F10.5,'  FINAL MASS = ',F10.5)
```

```
      END IF
C
C        *** IF MOMENTUM IS NOT CONSERVED WRITE AN ERROR MESSAGE          ***
C
      ERR=(RMJ-RMJ1)/RMJ1
      IF(DABS(ERR).GT.5.D-2) THEN
         WRITE(3,80)
80       FORMAT(' ERROR IN TWOJET:  SINGULARITIES IN CHANNEL SOLUTION')
C         IER=1
         RETURN
      END IF
C
      IER=0
C
      RETURN
      END
```

```
      FUNCTION U(Y)
C
C******************************************************************************
C                                                                             *
C     FUNCTION U COMPUTES THE JET VELOCITY                                    *
C                                                                             *
C   *** LATEST REVISION - 24 APR 1987 ***                                     *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C     INPUT:                                                                  *
C Y      - DISTANCE FROM THE JET CENTERLINE                                   *
C     OUTPUT:                                                                 *
C U      - HORIZONTAL COMPONENT OF VELOCITY                                   *
C                                                                             *
C******************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON U0,U1,P,A,B,Y1,ALP
C
      ARG1=ALP*(Y+Y1)/B
      ARG2=ALP*(Y-Y1)/B
      T1=DTANH(ARG1)
      T2=DTANH(ARG2)
      S1SQ=1.0D0-T1**2
      S2SQ=1.0D0-T2**2
C
      U=U0+A/2.0D0*(T1-T2)+U1*(S1SQ+S2SQ)
C
      RETURN
      END
```

```
      SUBROUTINE UPPVLC(X,U,UDOT)
C
C******************************************************************************
C                                                                             *
C     SUBROUTINE UPPVLC COMPUTES THE HORIZONTAL COMPONENT OF VELOCITY ON THE  *
C UPPER SIDE OF THE JET BOUNDARY.                                             *
C                                                                             *
C   *** LATEST REVISION - 24 APR 1987 ***                                     *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C     INPUT:                                                                  *
C X     - DISTANCE FROM THE JET ORIGIN                                        *
C     OUTPUT                                                                  *
C U     - HORIZONTAL COMPONENT OF VELOCITY AT THE STATION X                   *
C UDOT  - dU/dx AT THE STATION X                                              *
C                                                                             *
C******************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C        *** AREA16 IS SHARED WITH JETMAT, VLCJET, AND ONEJET ***
C
      COMMON /AREA16/ XU(100),YU(100),UU(100),VU(100),SPLNUU(100,3),NU
C
      CALL INTRP(X,XU,UU,NU,SPLNUU,100,U,UDOT,D2UDX2,IER)
      IF(IER.NE.0) THEN
         WRITE(3,10) IER,X
10       FORMAT(' ERROR IN UPPVLC:  INTRP RETURNED WITH IER = ',I3,
     1          ' X = ',F10.5)
         STOP
      END IF
C
      RETURN
      END
```

```
      FUNCTION USQ(Y)
C
C**************************************************************************
C                                                                         *
C     FUNCTION USQ COMPUTES THE SQUARE OF THE JET VELOCITY                *
C                                                                         *
C   *** LATEST REVISION - 24 APR 1987 ***                                 *
C                                                                         *
C   *** PARAMETER DESCRIPTION ***                                         *
C     INPUT:                                                              *
C Y      - DISTANCE FROM THE JET CENTERLINE                               *
C     OUTPUT:                                                             *
C USQ    - SQUARE OF THE JET VELOCITY                                     *
C                                                                         *
C**************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON U0,U1,P,A,B,Y1,ALP
C
      C=U(Y)
      USQ=C*C
C
      RETURN
      END
```

```
      FUNCTION V(RD,Y)
C
C***************************************************************************
C                                                                         *
C     FUNCTION V COMPUTES THE VERTICAL COMPONENT OF THE JET VELOCITY      *
C                                                                         *
C   *** LATEST REVISION - 24 APR 1987 ***                                 *
C                                                                         *
C   *** PARAMETER DESCRIPTION ***                                         *
C     INPUT:                                                              *
C RD     - VECTOR CONTAINING THE DERIVATIVES OF THE JET PARAMETERS        *
C Y      - DISTANCE FROM THE JET CENTERLINE                               *
C     OUTPUT:                                                             *
C V      - VERTICAL COMPONENT OF VELOCITY                                 *
C                                                                         *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RD(6)
      COMMON UO,U1,P,A,B,Y1,ALP
C
      CONST=ALP/B
      ETA=CONST*(Y-Y1)
      T=DTANH(ETA)
      DLC=DLOG(DCOSH(ETA))
      UODOT=RD(1)
      U1DOT=RD(2)
      ADOT=RD(4)
      BDOT=RD(5)
C
      V=-ETA/CONST*UODOT
     1  -T/CONST*U1DOT
     2  +0.5DO/CONST*(DLC-ETA)*ADOT
     3  +1.ODO/ALP*(-U1*ETA*T**2-(U1+A/2.ODO*ETA)*T+0.5DO*A*DLC+
     4                U1*ETA)*BDOT
C
      RETURN
      END
```

```
      SUBROUTINE VLCJET(ALWR,BLWR,AUPP,BUPP,Q,NJLS,NJLF,NJUS,NJUF,
     1                  N,VO,BETA,PATM)
C
C***************************************************************************
C                                                                          *
C     SUBROUTINE VLCJET COMPUTES VALUES OF THE VELOCITY COMPONENTS AT THE  *
C LOWER AND UPPER SIDES OF THE JET BOUNDARY.                               *
C SPLINE FITS ARE MADE TO THE VELOCITY COMPONENTS AND THE RESULTS SENT TO  *
C SUBROUTINES UPPVLC AND LWRVLC VIA COMMON BLOCKS.                         *
C                                                                          *
C    LATEST REVISION 24 APR 1987                                           *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C     INPUT:                                                               *
C  ALWR  - INFLUENCE COEFFICIENTS FOR U-VELOCITY ALONG THE LOWER BOUNDARY  *
C  BLWR  - INFLUENCE COEFFICIENTS FOR V-VELOCITY ALONG THE LOWER BOUNDARY  *
C  AUPP  - INFLUENCE COEFFICIENTS FOR U-VELOCITY ALONG THE LOWER BOUNDARY  *
C  BUPP  - INFLUENCE COEFFICIENTS FOR V-VELOCITY ALONG THE LOWER BOUNDARY  *
C  Q     - VECTOR OF SOURCE STRENGTHS                                      *
C  NJLS  - PANEL NUMBER OF THE BEGINNING OF THE LOWER JET BOUNDARY         *
C  NJLF  - PANEL NUMBER OF THE END OF THE LOWER JET BOUNDARY               *
C  NJUS  - PANEL NUMBER OF THE BEGINNING OF THE UPPER JET BOUNDARY         *
C  NJUF  - PANEL NUMBER OF THE END OF THE UPPER JET BOUNDARY               *
C  N     - NUMBER OF PANELS                                                *
C  VO    - FREE STREAM SPEED                                               *
C  BETA  - ANGLE OF ATTACK                                                 *
C                                                                          *
C     OUTPUT:                                                              *
C  PATM  - UPSTREAM AMBIENT PRESSURE                                       *
C                                                                          *
C     SENT VIA COMMON BLOCK IN AREA15                                      *
C  XL     - VECTOR CONTAINING THE ABSCISSA OF THE STATIONS AT WHICH THE    *
C           VELOCITIES ARE CALCULATED                                      *
C  YL     - VECTOR CONTAINING THE ORDINATES OF THE STATIONS AT WHICH THE   *
C           VELOCITIES ARE CALCULATED                                      *
C  UL     - VECTOR CONTAINING THE HORIZONTAL COMPONENT OF VELOCITY         *
C  VL     - VECTOR CONTAINING THE VERTICAL COMPONENT OF VELOCITY           *
C  SPLNUL- SPLINE FIT PARAMETERS FOR THE U COMPONENT OF VELOCITY AT THE JET *
C           LOWER BOUNDARY                                                 *
C                                                                          *
C    SENT VIA COMMON BLOCK IN AREA16                                       *
C  XU     - VECTOR CONTAINING THE ABSCISSA OF THE STATIONS AT WHICH THE    *
C           VELOCITIES ARE CALCULATED                                      *
C  YU     - VECTOR CONTAINING THE ORDINATES OF THE STATIONS AT WHICH THE   *
C           VELOCITIES ARE CALCULATED                                      *
C  UU     - VECTOR CONTAINING THE HORIZONTAL COMPONENT OF VELOCITY         *
C  VU     - VECTOR CONTAINING THE VERTICAL COMPONENT OF VELOCITY           *
C  SPLNUU- SPLINE FIT PARAMETERS FOR THE U COMPONENT OF VELOCITY AT THE JET *
C           LOWER BOUNDARY                                                 *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALWR(NJLF-NJLS+1,N),BLWR(NJLF-NJLS+1,N),
     1          AUPP(NJUF-NJUS+1,N),BUPP(NJUF-NJUS+1,N),Q(N)
C
C        *** AREA15 IS SHARED WITH JETMAT, LWRVLC, AND ONEJET           ***
C        *** AREA16 IS SHARED WITH JETMAT, UPPVLC, AND ONEJET           ***
C
      COMMON /AREA15/ XL(100),YL(100),UL(100),VL(100),SPLNUL(100,3),NL
```

```
      COMMON /AREA16/ XU(100),YU(100),UU(100),VU(100),SPLNUU(100,3),NU
C
C        *** CALCULATE AND STORE VELOCITY COMPONENTS FOR THE LOWER ***
C        *** JET BOUNDARY                                          ***
C
      NL=NJLF-NJLS+1
      DO I=NJLF,NJLS,-1
         II=NJLF-I+1
         SUM1=0.0D0
         SUM2=0.0D0
         DO J=1,N
            SUM1=SUM1+ALWR(II,J)*Q(J)
            SUM2=SUM2+BLWR(II,J)*Q(J)
         END DO
         UL(II)=V0*DCOS(BETA)+SUM1
         VL(II)=V0*DSIN(BETA)+SUM2
      END DO
C
C        *** SPLINE FIT THE LOWER VELOCITY COMPONENTS ***
C
      CALL IQHSCU(XL,UL,NL,SPLNUL,100,IER)
      IF(IER.NE.0) THEN
         WRITE(5,50) IER
         STOP
      END IF
C
C        *** CALCULATE AND STORE VELOCITY COMPONENTS FOR THE UPPER ***
C        *** JET BOUNDARY                                          ***
C
      NU=NJUF-NJUS+1
      DO I=NJUS,NJUF
         II=I-NJUS+1
         SUM1=0.0D0
         SUM2=0.0D0
         DO J=1,N
            SUM1=SUM1+AUPP(II,J)*Q(J)
            SUM2=SUM2+BUPP(II,J)*Q(J)
         END DO
         UU(II)=V0*DCOS(BETA)+SUM1
         VU(II)=V0*DSIN(BETA)+SUM2
      END DO
C
C        *** SPLINE FIT THE UPPER VELOCITY COMPONENTS ***
C
      CALL IQHSCU(XU,UU,NU,SPLNUU,100,IER)
      IF(IER.NE.0) THEN
         WRITE(5,50) IER
         STOP
      END IF
C
C        *** CALCULATE THE UPSTREAM ATMOSPHERIC PRESSURE ***
C
      PATM=0.5D0*(UU(2)**2-V0**2)
C
50    FORMAT(' ERROR IN JETVLC, CALL TO IQHSCU RETURNED WITH '
     1       'IER = ',I5)
C
      RETURN
      END
```

B.3.3 Panel Method Library PAN2LIB

```
      SUBROUTINE GEOM(XBOD,YBOD,ZETA,CX,CY,WORK,N,XCP,YCP,ALPHA,D,
     1                IND1,IND2,PD,PE,PF,PG,PH,PPI,C)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE GEOM COMPUTES THE SURFACE ELEMENT LENGTH, RADIUS OF         *
C CURVATURE, ORIENTATION IN SPACE, AND PARABOLIC FIT COEFFICIENTS            *
C                                                                            *
C   *** LATEST REVISION - 28 JAN 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  XBOD   - VECTOR OF BODY X COORDINATES                                     *
C  YBOD   - VECTOR OF BODY Y COORDINATES                                     *
C  ZETA   - WORK SPACE VECTOR FOR THE SPLINE FIT                             *
C  CX     - WORK SPACE MATRIX FOR THE X SPLINE FIT COEFFICIENTS              *
C  CY     - WORK SPACE MATRIX FOR THE Y SPLINE FIT COEFFICIENTS              *
C  WORK   - WORK SPACE MATRIX FOR PERIODIC SPLINE FITS                       *
C  N      - NUMBER OF SURFACE ELEMENTS                                       *
C                                                                            *
C     OUTPUT:                                                                *
C  XCP    - VECTOR OF CONTROL POINT X COORDINATES                            *
C  YCP    - VECTOR OF CONTROL POINT Y COORDINATES                            *
C  ALPHA  - VECTOR OF INVERSE TANGENTS OF THE SLOPE OF EACH PANEL            *
C           (ORIENTATION ANGLE)                                              *
C  D      - VECTOR CONTAINING THE LENGTHS OF EACH PANE                       *
C  IND1   - VECTOR OF INDEX OF THE PANEL WHICH ADJOINS TO THE LEFT           *
C  IND2   - VECTOR OF INDEX OF THE PANEL WHICH ADJOINS TO THE RIGHT          *
C  PD..PPI- PARABOLIC FIT COEFFICIENTS                                       *
C  C      - VECTOR OF SURFACE CURVATURE COEFFICIENTS                         *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL PERDT
      DIMENSION XBOD(N+1),YBOD(N+1),ZETA(N+1),CX(N,3),CY(N,3),
     1          WORK(6*(N+1)),XCP(N),YCP(N),ALPHA(N),D(N),IND1(N),
     2          IND2(N),PD(N),PE(N),PF(N),PG(N),PH(N),PPI(N),C(N)
C
      PI=3.141592654D0
C
C        *** CHECK FOR PERIODIC GEOMETRY ***
C
      XDIFF=XBOD(N+1)-XBOD(1)
      YDIFF=YBOD(N+1)-YBOD(1)
      IF(DABS(XDIFF).LT.1.E-3.AND.DABS(YDIFF).LT.1.E-3) THEN
         PERDT=.TRUE.
         XBOD(N+1)=XBOD(1)
         YBOD(N+1)=YBOD(1)
      ELSE
         PERDT=.FALSE.
      END IF
C
       DO I=1,N
C
         DX=XBOD(I+1)-XBOD(I)
         DY=YBOD(I+1)-YBOD(I)
C
```

```
C          *** COMPUTE THE PANEL LENGTH ***
C
         D(I)=DSQRT(DX**2+DY**2)
C
C          *** COMPUTE THE PANEL ORIENTATION ANGLE ***
C
         IF(DABS(DX).LT.1.D-6) THEN
            IF(DY.GT.0.0D0) THEN
               ALPHA(I)=PI/2.0D0
            ELSE
               ALPHA(I)=-PI/2.0D0
            END IF
         ELSE
            ALPHA(I)=DATAN(DY/DX)
            IF(DY.LT.0.0D0.AND.DX.LT.0.0D0) ALPHA(I)=ALPHA(I)-PI
            IF(DY.GE.0.0D0.AND.DX.LT.0.0D0) ALPHA(I)=ALPHA(I)+PI
         END IF
C
      END DO
C
C        *** SPLINE FIT THE BODY COORDINATES AS A FUNCTION OF THE ***
C        *** PANEL LENGTH                                         ***
C
      ZETA(1)=0.0D0
      DO I=2,N+1
         ZETA(I)=ZETA(I-1)+D(I-1)
      END DO
C
      IF(PERDT) THEN
         CALL ICSPLN(ZETA,XBOD,N+1,CX,N,WORK,IER)
         IF(IER.NE.0) THEN
            WRITE(3,7) IER
7           FORMAT(' ERROR IN GEOM, ICSCCU RETURNED WITH IER = ',I4)
            STOP
         END IF
         CALL ICSPLN(ZETA,YBOD,N+1,CY,N,WORK,IER)
         IF(IER.NE.0) THEN
            WRITE(3,7) IER
            STOP
         END IF
      ELSE
         CALL ICSCCU(ZETA,XBOD,N+1,CX,N,IER)
         IF(IER.NE.0) THEN
            WRITE(3,7) IER
            STOP
         END IF
         CALL ICSCCU(ZETA,YBOD,N+1,CY,N,IER)
         IF(IER.NE.0) THEN
            WRITE(3,7) IER
            STOP
         END IF
      END IF
C
      DO I=1,N
C
C          *** FIND THE CONTROL POINT LOCATION AND SURFACE DERIVATIVES ***
C
         Z=0.5D0*(ZETA(I)+ZETA(I+1))
         CALL INTRP(Z,ZETA,XBOD,N+1,CX,N,XX,DXDZ,D2XDZ2,IER)
         IF(IER.NE.0) THEN
            WRITE(3,20) IER,Z
```

```
20          FORMAT(' ERROR IN GEOM: INTRP RETURNED WITH IER = ',I3,
     1             ' X = ',F10.5)
            STOP
         END IF
         CALL INTRP(Z,ZETA,YBOD,N+1,CY,N,YY,DYDZ,D2YDZ2,IER)
         IF(IER.NE.0) THEN
            WRITE(3,20) IER,Z
            STOP
         END IF
         XCP(I)=XX
         YCP(I)=YY
         C(I)=(DXDZ*D2YDZ2-DYDZ*D2XDZ2)/(DXDZ**2+DYDZ**2)**1.5D0
C
C           *** COMPUTE PARABOLIC FIT COEFFICIENTS ***
C
         L1=I-1
         L2=I
         L3=I+1
         IF(PERDT.AND.I.EQ.1) L1=N
         IF(PERDT.AND.I.EQ.N) L3=1
         IF(.NOT.PERDT.AND.(I.EQ.1.OR.I.EQ.N)) THEN
            PD(L2)=0.0D0
            PE(L2)=0.0D0
            PF(L2)=0.0D0
            PG(L2)=0.0D0
            PH(L2)=0.0D0
            PPI(L2)=0.0D0
         ELSE
            A=0.5D0*(D(L1)+D(L2))
            B=0.5D0*(D(L2)+D(L3))
            PD(L2)=-B/(A*(A+B))
            PE(L2)=(B-A)/(A*B)
            PF(L2)=A/(B*(A+B))
            PG(L2)=2.0D0/(A*(A+B))
            PH(L2)=-2.0D0/(A*B)
            PPI(L2)=2.0D0/(B*(A+B))
         END IF
C
      END DO
C
C        *** FILL THE INDEX ARRAYS ***
C
      DO I=1,N
         IND1(I)=I-1
         IND2(I)=I+1
      END DO
C
      IF(PERDT) THEN
         IND1(1)=N
         IND2(N)=1
      ELSE
         IND2(N)=0
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE GETDAT(NUNIT,XBOD,YBOD,VN,N,VO,BETA)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE GETDAT READS THE DATA FILE BODY.DAT TO OBTAIN THE COORDINATES *
C OF THE SHROUD GEOMETRY AS WELL AS THE NORMAL VELOCITY AT EACH PANEL.       *
C                                                                            *
C   *** LATEST REVISION - 28 JAN 1987 ***                                    *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  NUNIT  - LOGICAL UNIT FOR DATA INPUT                                      *
C                                                                            *
C  THE INPUT IS THE DATA FILE BODY.DAT WHICH CONTAINS THE X AND Y COORDINATES *
C  ALONG WITH THE TRANSPIRATION VELOCITY FOR EACH PANEL                      *
C                                                                            *
C     OUTPUT:                                                                *
C  XBOD  - VECTOR CONTAINING THE ABSCISSA OF THE BODY POINTS                 *
C  YBOD  - VECTOR CONTAINING THE ORDINATES OF THE BODY POINTS                *
C  VN    - VECTOR OF PANEL TRANSPIRATION VELOCITIES                          *
C  N     - NUMBER OF SURFACE ELEMENTS (NUMBER OF BODY POINTS - 1)            *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XBOD(1),YBOD(1),VN(1)
C
      REWIND NUNIT
      REWIND 2
C
      DO I=1,500
         READ(NUNIT,*,END=10) XBOD(I),YBOD(I),VN(I)
      END DO
10    N=I-2
C
      READ(2,*) VO,BETA
C
      RETURN
      END
```

```
      SUBROUTINE INFINV(XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,PF,PG,PH,PPI,C,
     1                  WORK,A,B,W,N,AMAT,BMAT,WINV)
C
C***************************************************************************
C                                                                          *
C     SUBROUTINE INFINV COMPUTES THE INVERSE OF THE AERODYNAMIC COEFFICIENT *
C MATRIX.                                                                  *
C                                                                          *
C   *** LATEST REVISION - 28 JAN 1987 ***                                  *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C     INPUT:                                                               *
C  XCP   - VECTOR CONTAINING THE CONTROL POINT X COORDINATES               *
C  YCP   - VECTOR CONTAINING THE CONTROL POINT Y COORDINATES               *
C  ALPHA - VECTOR CONTAINING THE SURFACE SLOPE ANGLES FOR EACH PANEL       *
C  D     - VECTOR CONTAINING THE PANEL LENGTHS                             *
C  IND1  - VECTOR OF PANEL INDEX WHICH ADJOINS TO THE LEFT                 *
C  IND2  - VECTOR OF PANEL INDEX WHICH ADJOINS TO THE RIGHT                *
C  PD..PPI PARABOLIC FIT COEFFICIENTS                                      *
C  C     - VECTOR OF SURFACE CURVATURE COEFFICIENTS                        *
C  WORK  - WORK SPACE MATRIX                                               *
C  A     - WORK SPACE VECTOR TO HOLD X VELOCITY INFLUENCE COEFFICIENTS     *
C  B     - WORK SPACE VECTOR TO HOLD Y VELOCITY INFLUENCE COEFFICIENTS     *
C  W     - WORK SPACE MATRIX TO TEMPORARILY HOLD THE INFLUENCE COEFFICIENTS *
C  N     - NUMBER OF PANELS                                                *
C                                                                          *
C     OUTPUT:                                                              *
C  AMAT  - MATRIX IF HORIZONTAL INDUCED VELOCITIES                         *
C  BMAT  - MATRIX OF VERTICAL INDUCED VELOCITIES                           *
C  WINV  - INVERSE OF THE AERODYNAMIC INFLUENCE COEFFICIENT MATRIX         *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XCP(N),YCP(N),ALPHA(N),D(N),PD(N),PE(N),PF(N),PG(N),
     1          PH(N),PPI(N),C(N),WINV(N,N),W(N,N),WORK(8*N),IND1(N),
     2          IND2(N),A(N),B(N),AMAT(N,N),BMAT(N,N)
C
C        *** GENERATE THE AERODYNAMIC INFLUENCE COEFFICIENT MATRIX ***
C
      DO I=1,N
         X=XCP(I)
         Y=YCP(I)
         CALL INFLCE(X,Y,XCP,YCP,ALPHA,D,IND1,IND2,PD,PE,PF,PG,PH,
     1               PPI,C,WORK,N,A,B)
         DO J=1,N
            AMAT(I,J)=A(J)
            BMAT(I,J)=B(J)
            W(I,J)=B(J)*DCOS(ALPHA(I))-A(J)*DSIN(ALPHA(I))
         END DO
      END DO
C
C        *** INVERT THE MATRIX USING LINV1F ***
C
      CALL LINV1F(W,N,N,WINV,0,WORK,IER)
      IF(IER.EQ.129) THEN
         WRITE(3,20)
20       FORMAT(' ERROR IN INFINV, LINV1F FOUND A SINGULAR MATRIX ')
         STOP
      END IF
```

```
C
      RETURN
      END
```

```
      SUBROUTINE INFLCE(X,Y,XCP,YCP,ALPHA,D,IND1,IND2,
     1                  PD,PE,PF,PG,PH,PPI,CC,W,N,A,B)
C
C***************************************************************************
C                                                                          *
C     SUBROUTINE INFLCE COMPUTES THE AERODYNAMIC INFLUENCE COEFFICIENTS FOR *
C USE IN THE HIGHER ORDER PANEL METHOD.                                    *
C                                                                          *
C   *** LATEST REVISION - 28 JAN 1987 ***                                  *
C                                                                          *
C   *** PARAMETER DESCRIPTION ***                                          *
C                                                                          *
C     INPUT:                                                               *
C  X      - X COORDINATE AT WHICH THE INFLUENCE COEFFICIENT IS TO BE CALCULATED *
C  Y      - Y COORDINATE AT WHICH THE INFLUENCE COEFFICIENT IS TO BE CALCULATED *
C  XCP    - VECTOR OF CONTROL POINT X COORDINATES                          *
C  YCP    - VECTOR OF CONTROL POINT Y COORDINATES                          *
C  ALPHA  - VECTOR OF SURFACE SLOPES FOR EACH PANEL                        *
C  D      - VECTOR OF PANEL LENGTHS                                        *
C  IND1   - VECTOR OF PANEL INDEX WHICH ADJOINS TO THE LEFT                *
C  IND2   - VECTOR OF PANEL INDEX WHICH ADJOINS TO THE RIGHT               *
C  PD..PPI PARABOLIC FIT COEFFICIENTS                                      *
C  CC     - VECTOR OF SURFACE CURVATURE COEFFICIENTS                       *
C  W      - WORK SPACE FOR TEMPORARILY STORING THE INDUCED VELOCITY COMPONENTS *
C  N      - NUMBER OF PANELS                                               *
C                                                                          *
C     OUTPUT:                                                              *
C  A      - VECTOR OF INFLUENCE COEFFICIENTS FOR THE X COMPONENT OF VELOCITY *
C  B      - VECTOR OF INFLUENCE COEFFICIENTS FOR THE Y COMPONENT OF VELOCITY *
C                                                                          *
C***************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XCP(N),YCP(N),ALPHA(N),D(N),PD(N),PE(N),PF(N),PG(N),
     1          PH(N),PPI(N),CC(N),IND1(N),IND2(N),W(8,N),A(N),B(N)
C
      PI=3.14159265D0
C
      DO J=1,N
C
         C=DCOS(ALPHA(J))
         S=DSIN(ALPHA(J))
C
         RX=X-XCP(J)
         RY=Y-YCP(J)
         RO=DSQRT(RX**2+RY**2)
         IF(RO.EQ.0) THEN
            EPS=1.D8
         ELSE
            EPS=D(J)/RO
         END IF
         EPS2=EPS**2
C
         IF(EPS.LT.7.5D-2) THEN
C
C              *** USE APPROXIMATE FORMULAS IF THE FIELD POINT IS VERY FAR ***
C              *** AWAY FROM THE PANEL CONTROL POINT                       ***
C
            ALP=RX/RO
            BET=RY/RO
            AUX1=ALP*C+BET*S
```

```
          AUX2=-ALP*S+BET*C
C
          W(1,J)=2.0D0*EPS*ALP
          W(2,J)=2.0D0*EPS*BET
          W(3,J)=EPS2/6.0D0*(2.0D0*ALP*AUX1-C)
          W(4,J)=EPS2/6.0D0*(2.0D0*BET*AUX1-S)
          W(5,J)=EPS2/12.0D0*(2.0D0*ALP*AUX2+S)
          W(6,J)=EPS2/12.0D0*(2.0D0*BET*AUX2-C)
          W(7,J)=W(1,J)/24.0D0
          W(8,J)=W(2,J)/24.0D0
C
        ELSE
C
          XI=RX*C+RY*S
          ETA=-RX*S+RY*C
C
          IF(EPS.LT.3.0D-1) THEN
C
C            *** USE ANOTHER SET OF APPROXIMATIONS IF THE FIELD POINT IS ***
C            *** MODERATELY FAR AWAY FROM THE PANEL CONTROL POINT        ***
C
             ALP=XI/RO
             ALP2=ALP**2
             BET=ETA/RO
             BET2=BET**2
             AUX1=(ALP2/3.0D0-0.25D0)*EPS2
             AUX2=(ALP2/3.0D0-1.0D0/12.0D0)*EPS2
             AUX3=2.0D0*ALP2-1.0D0
             AUX4=(8.0D0*(ALP2-1.0D0)*ALP2+1.0D0)*EPS2
C
             VOX=2.0D0*ALP*EPS*(1.0D0+AUX1)
             VOE=2.0D0*BET*EPS*(1.0D0+AUX2)
             V1X=EPS2/6.0D0*(AUX3+1.5D-1*AUX4)
             V1E=ALP*BET*EPS2/3.0D0*(1.0D0+0.3D0*AUX3*EPS2)
             VCX=ALP*BET*EPS2/6.0D0*(1.0D0+0.9D0*AUX3*EPS2)
             VCE=EPS2/12.0D0*((2.0D0*BET2-1.0D0)-7.5D-2*AUX4)
             V2X=ALP*EPS/12.0D0*(1.0D0+1.8D0*AUX1)
             V2E=BET*EPS/12.0D0*(1.0D0+1.8D0*AUX2)
C
          ELSE
C
C               *** USE THE EQUATIONS WITHOUT APPROXIMATION IF THE FIELD ***
C               *** POINT IS CLOSE TO THE PANEL CONTROL POINT            ***
C
             XI2=XI**2
             ETA2=ETA**2
             DEL=D(J)
             DEL2=DEL**2
             R1SQ=(XI+0.5D0*DEL)**2+ETA2
             R2SQ=(XI-0.5D0*DEL)**2+ETA2
             C1=DLOG(R1SQ/R2SQ)
             RNUM=ETA*DEL
             DENOM=XI2+ETA2-0.25D0*DEL2
             C2=2.0D0*DATAN(RNUM/DENOM)
             IF(DABS(RNUM).LT.1.D-6.AND.DENOM.LT.0.0D0) THEN
                C2=2.0D0*PI
             ELSE
                IF(RNUM.GT.0.0D0.AND.DENOM.LT.0.0D0) C2=C2+2.0D0*PI
                IF(RNUM.LT.0.0D0.AND.DENOM.LT.0.0D0) C2=C2-2.0D0*PI
             END IF
             AUX1=ETA*C2+XI*C1
```

```
            AUX2=XI*C2-ETA*C1
C
            VOX=C1
            VOE=C2
            V1X=(AUX1-2.0D0*DEL)/DEL
            V1E=AUX2/DEL
            VCX=(-AUX2+0.5D0*XI*ETA*(DEL**3)/(R1SQ*R2SQ))/DEL
            VCE=(AUX1-DEL*(1.0D0+((XI2+ETA2)**2-
     1                          0.25D0*(XI2-ETA2)*DEL2)/
     2                         (R1SQ*R2SQ)))/DEL
            V2X=(XI*ETA*C2+0.5D0*(XI2-ETA2)*C1-XI*DEL)/DEL2
            V2E=(0.5D0*(XI2-ETA2)*C2-XI*ETA*C1+ETA*DEL)/DEL2
C
          END IF
C
C           *** TRANSFORM TO THE GLOBAL COORDINATE SYSTEM ***
C
          W(1,J)=VOX*C-VOE*S
          W(2,J)=VOX*S+VOE*C
          W(3,J)=V1X*C-V1E*S
          W(4,J)=V1X*S+V1E*C
          W(5,J)=VCX*C-VCE*S
          W(6,J)=VCX*S+VCE*C
          W(7,J)=V2X*C-V2E*S
          W(8,J)=V2X*S+V2E*C
C
        END IF
C
      END DO
C
C
C       *** COMPUTE THE INFLUENCE COEFFICIENTS ***
C
      DO J=1,N
C
        JM1=IND1(J)
        JP1=IND2(J)
C
        VXJ  =W(1,J)+W(3,J)*PE(J)*D(J)+W(5,J)*CC(J)*D(J)+
     1        W(7,J)*(PH(J)+CC(J)**2)*D(J)**2
        VYJ  =W(2,J)+W(4,J)*PE(J)*D(J)+W(6,J)*CC(J)*D(J)+
     1        W(8,J)*(PH(J)+CC(J)**2)*D(J)**2
        IF(JM1.NE.0) THEN
           VXJM1=W(3,JM1)*PF(JM1)*D(JM1)+W(7,JM1)*PPI(JM1)*D(JM1)**2
           VYJM1=W(4,JM1)*PF(JM1)*D(JM1)+W(8,JM1)*PPI(JM1)*D(JM1)**2
        ELSE
           VXJM1=0.0D0
           VYJM1=0.0D0
        END IF
        IF(JP1.NE.0) THEN
           VXJP1=W(3,JP1)*PD(JP1)*D(JP1)+W(7,JP1)*PG(JP1)*D(JP1)**2
           VYJP1=W(4,JP1)*PD(JP1)*D(JP1)+W(8,JP1)*PG(JP1)*D(JP1)**2
        ELSE
           VXJP1=0.0D0
           VYJP1=0.0D0
        END IF
C
        A(J)=VXJ+VXJM1+VXJP1
        B(J)=VYJ+VYJM1+VYJP1
C
      END DO
```

```
C
      RETURN
      END
```

```
      SUBROUTINE STRNTH(ALPHA,VN,WINV,N,V0,BETA,Q)
C
C***********************************************************************
C                                                                      *
C     SUBROUTINE STRNTH COMPUTES THE PANEL SOURCE STRENGTHS.           *
C                                                                      *
C   *** LATEST REVISION - 28 JAN 1987 ***                              *
C                                                                      *
C   *** PARAMETER DESCRIPTION ***                                      *
C                                                                      *
C     INPUT:                                                           *
C  ALPHA - VECTOR CONTAINING THE SURFACE SLOPE FOR EACH PANEL          *
C  VN    - VECTOR CONTAINING THE TRANSPIRATION VELOCITY FOR EACH PANEL *
C  WINV  - INVERSE OF THE AERODYNAMIC INFLUENCE COEFFICIENTS           *
C  N     - NUMBER OF PANELS                                            *
C  V0    - FREE STREAM VELOCITY                                        *
C  BETA  - ANGLE OF ATTACK                                             *
C                                                                      *
C     OUTPUT:                                                          *
C  Q     - VECTOR CONTAINING THE SOURCE STRENGTHS                      *
C                                                                      *
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALPHA(N),VN(N),WINV(N,N),Q(N)
C
      DO I=1,N
         SUM=0.0D0
         DO J=1,N
            SUM=SUM+WINV(I,J)*(V0*DSIN(ALPHA(J)-BETA)-VN(J))
         END DO
         Q(I)=SUM
      END DO
C
      RETURN
      END
```

B.3.4 Mathematics library MATHLIB

```
      SUBROUTINE INTRP(X,XVEC,YVEC,N,SPLN,NS,Y,DYDX,D2YDX2,IER)
C
C***********************************************************************
C                                                                      *
C    THIS SUBROUTINE USES CUBIC SPLINE FIT PARAMETERS PRODUCED BY IMSL *
C ROUTINE ICSCCU TO FIND INTERPOLATED VALUES OF A FUNCTION AND ITS DERIVATIVES*
C AT ANY STATION X.                                                    *
C                                                                      *
C   *** PARAMETER DESCRIPTION ***                                      *
C                                                                      *
C     INPUT:                                                           *
C  X    - INDEPENDENT COORDINATE.  X MUST BE WITHIN THE RANGE OF WHICH WAS *
C         SENT TO SUBROUTINE SPLINE.                                   *
C  XVEC - VECTOR OF LENGTH N CONTAINING THE X COORDINATES.             *
C  YVEC - VECTOR OF LENGTH N CONTAINING THE  VALUE OF Y AT X STATIONS  *
C         CORRESPONDING TO THOSE IN XVEC.                              *
C  N    - NUMBER OF DATA POINTS USED IN THE SPLINE FIT (DIMENSION OF VECTORS *
C         XVEC AND YVEC)                                               *
C  SPLN - VECTOR OF SPLINE FIT PARAMETERS AS OBTAINED FROM A CALL TO SYSTEM *
C         ROUTINE ICSCCU.                                              *
C  NS   - ROW DIMENSION OF SPLN EXACTLY AS SPECIFIED IN THE DIMENSION  *
C         STATEMENT OF THE CALLING ROUTINE                             *
C                                                                      *
C     OUTPUTS                                                          *
C  Y    - INTERPOLATED VALUE OF THE FUNCTION AT THE STATION X          *
C  DYDX - INTERPOLATED VALUE OF THE FIRST DERIVATIVE OF THE FUNCTION AT THE *
C         STATION X                                                    *
C  D2YDX2 INTERPOLATED VALUE OF THE SECOND DERIVATIVE AT THE STATION X *
C  IER  - ERROR PARAMETER.  0 FOR SUCCESSFUL INTERPOLATION             *
C                           1 FOR X OUT OF BOUNDS                      *
C                                                                      *
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XVEC(N),YVEC(N),SPLN(NS,3)
C
C        *** VERIFY THAT X IS WITHIN THE PROPER RANGE ***
C        *** EPS IS USED AS A TOLERANCE FOR ROUND-OFF ERROR ***
C
      EPS=1.0D-6
      IF((XVEC(1)-X).GT.EPS.OR.(X-XVEC(N)).GT.EPS) THEN
         IER=1
         RETURN
      END IF
C
C        *** SEARCH THROUGH THE ABSCISSA VECTOR TO LOCATE THE INTERVAL IN ***
C        *** WHICH X LIES.                                                ***
C
      NF=N-1
      DO 10 J=1,NF
         IF(J.EQ.NF) GOTO 20
         IF((XVEC(J)-EPS).LE.X.AND.X.LT.XVEC(J+1)) GOTO 20
10    CONTINUE
C
C        *** COMPUTE INTERPOLATED VALUES ***
C
20    D=X-XVEC(J)
      Y=SPLN(J,3)*D**3+SPLN(J,2)*D**2+SPLN(J,1)*D+YVEC(J)
      DYDX=3.0D0*SPLN(J,3)*D**2+2.0D0*SPLN(J,2)*D+SPLN(J,1)
```

```
      D2YDX2=6.0D0*SPLN(J,3)*D+2.0D0*SPLN(J,2)
C
      IER=0
C
      RETURN
      END
```

```
      SUBROUTINE LINTRP(X,XP,YP,SPLN,N,Y,DYDX,IER)
C
C****************************************************************************
C                                                                           *
C    SUBROUTINE LINTRP WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS  *
C AND ACOUSTICS AT STANFORD UNIVERSITY BY THOMAS LUND.  LATEST REVISION 17  *
C JULY 1984.                                                                *
C                                                                           *
C    THIS SUBROUTINE USES SLOPES GENERATED BY SUBROUTINE LNSPLN TO FIND     *
C INTERPOLATED VALUES OF A FUNCTION AND ITS DERIVATIVE AT ANY  STATION X.   *
C                                                                           *
C                                                                           *
C  **PARAMETER DESCRIPTION**                                                *
C                                                                           *
C     INPUTS:                                                               *
C  X  - INDEPENDENT COORDINATE.  X MUST BE WITHIN THE RANGE OF WHICH WAS    *
C        SENT TO SUBROUTINE LNSPLN.                                         *
C  XP - VECTOR OF LENGTH N CONTAINING THE X COORDINATES OF A FUNCTION P.    *
C  YP - VECTOR OF LENGTH N CONTAINING THE  VALUE OF P AT X STATIONS         *
C        CORRESPONDING TO THOSE IN XP.                                      *
C  SPLN- VECTOR OF SLOPES AS OBTAINED FROM A CALL TO SUBROUTINE LNSPLN.     *
C  N  - NUMBER OF DATA POINTS USED IN THE SPLINE FIT (DIMENSION OF VECTORS XP *
C        AND YP)                                                            *
C                                                                           *
C     OUTPUTS:                                                              *
C  Y  - INTERPOLATED VALUE OF THE FUNCTION AT THE STATION X                 *
C  DYDX - INTERPOLATED VALUE OF THE FIRST DERIVATIVE OF THE FUNCTION AT THE *
C        STATION X                                                          *
C  IER - ERROR PARAMETER, ON SUCCESSFUL TERMINATION IER IS SET TO ZERO, IER=1 *
C         INDICATES THAT X WAS OUT OF BOUNDS OF THE                         *
C         SPLINE FIT SLOPES.                                                *
C                                                                           *
C                                                                           *
C  **PRECISION** - ALL PARAMETERS AND INTERNAL VARIABLES ARE DOUBLE PRECISION *
C                                                                           *
C****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(N),YP(N),SPLN(N-1)
C
      IER=0
      NF=N-1
C
C         *** VERIFY THAT X IS WITHIN THE PROPER RANGE ***
C         *** EPS IS USED AS A TOLERANCE FOR ROUND-OFF ERROR ***
C
      EPS=1.0D-6
      IF((XP(1)-X).GT.EPS.OR.(X-XP(N)).GT.EPS) THEN
         IER=1
         RETURN
      END IF
C
C         *** SEARCH THROUGH THE ABSCISSA VECTOR TO LOCATE THE INTERVAL IN ***
C         *** WHICH X LIES. ***
C
      DO 10 J=1,NF
         IF(J.EQ.NF) GOTO 20
         IF(X.GE.(XP(J)-EPS).AND.X.LT.XP(J+1)) GOTO 20
10    CONTINUE
C
C         *** COMPUTE INTERPOLATED VALUES ***
```

```
C
20    D=X-XP(J)
      Y=SPLN(J)*D+YP(J)
      DYDX=SPLN(J)
C
      RETURN
      END
```

```
      SUBROUTINE LNSPLN(X,Y,N,SLOPE,IER)
C***********************************************************************
C                                                                      *
C     THIS SUBROUTINE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUT- *
C ICS AND ACOUSTICS, STANFORD UNIVERSITY BY THOMAS LUND.  LATEST REVIS-*
C ION 13 SEPTEMBER 1984.                                               *
C                                                                      *
C     SUBROUTINE LNSPLN (LINEAR SPLINE FIT) IS USED TO GENERATE THE    *
C SLOPE OF A DISCRETE FUNCTION THROUGH THE USE OF LINEAR SEGMENTS. THE *
C SLOPE AT THE MIDPOINT OF EACH INTERVAL IS COMPUTED USING FIRST       *
C ORDER ACCURATE BACKWARD DIFFERENCING.  SUBROUTINE LINTRP IS CALLED   *
C TO DO THE ACTUAL INTERPOLATING.                                      *
C                                                                      *
C  **PARAMETER DESCRIPTION**                                           *
C                                                                      *
C     INPUT:                                                           *
C     X -  VECTOR OF LENGTH N CONTAINING THE ABSCISSAS.  THE ELEMENTS OF*
C          X MUST BE ORDERED SUCH THAT X(I+1)>X(I).                    *
C     Y  - VECTOR OF LENGTH N CONTAINING THE ORDINATES.                *
C     N  - LENGTH OF THE INPUT VECTORS.  N MUST BE GREATER THAN ONE.   *
C                                                                      *
C     OUTPUT:                                                          *
C SLOPE - VECTOR OF LENGTH N-1 CONTAINING THE SLOPE OF EACH INTERVAL   *
C   IER - ERROR PARAMETER.  ON NORMAL EXIT IER IS SET TO ZERO.  IER=1  *
C          INDICATES THAT N WAS LESS THAN 2.  IER=2 INDICATES THAT     *
C          X(I+1)<=X(I).                                               *
C                                                                      *
C  **LINKING** - NO EXTERNAL SUBROUTINES TO LINK.                      *
C                                                                      *
C  **PRECISION** - ALL PARAMETERS AND INTERNAL VARIABLES ARE DOUBLE    *
C                   PRECISION.                                         *
C                                                                      *
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N),Y(N),SLOPE(N-1)
      NF=N-1
C     CHECK FOR ERROR CONDITIONS
      IER=0
      IF(N.LT.2) THEN
         IER=1
         GOTO 200
      END IF
      DO 10 I=1,NF
         IF(X(I+1).LE.X(I)) THEN
            IER=2
            GOTO 200
         END IF
10    CONTINUE
C     COMPUTE FIRST ORDER ACCURATE SLOPES
      DO 20 I=1,NF
         SLOPE(I)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
20    CONTINUE
200   RETURN
      END
```

```
      SUBROUTINE RK2(N,FCN,X,Y,XEND)
C**********************************************************************
C                                                                     *
C   THIS ROUTINE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS  *
C AND ACOUSTICS, STANFORD UNIVERSITY BY THOMAS LUND.  LATEST REVISION *
C 20 JAN 1985.                                                        *
C                                                                     *
C   SUBROUTINE RK2 INTEGRATES A FIRST ORDER SYSTEM OF ORDINARY DIFFER- *
C ENTIAL EQUATIONS USING A SECOND ORDER ACCURATE RUNGE-KUTTA SCHEME.  *
C EACH CALL TO RK2 ADVANCES THE SOLUTION FORWARD IN TIME ONE INTERVAL. *
C                                                                     *
C  ***PARAMETER DESCRIPTION***                                        *
C                                                                     *
C  N  -  RANK OF THE FIRST ORDER SYSTEM.                              *
C  FCN - N-DIMENSIONAL FUNCTION WHICH DEFINES THE SYSTEM DERIVATIVE.  *
C  X   - INDEPENDENT VARIABLE, INITIAL VALUE FOR INTEGRATION STEP.    *
C  Y   - VECTOR OF LENGTH N WHICH ON INPUT CONTAINS THE INITIAL VALUES *
C        AND ON OUTPUT CONTAINS THE APPROXIMATE SOLUTION ADVANCED IN  *
C        TIME ONE INTERVAL.                                           *
C XEND - VALUE OF THE INDEPENDENT VARIABLE AT THE END OF THE INTERVAL. *
C        THE INTERVAL SIZE IS DEFINED AS XEND-X.                      *
C                                                                     *
C  ***LINKING***                                                      *
C                                                                     *
C     NO LINKING TO EXTERNAL ROUTINES IS NECESSARY, BUT A DRIVING     *
C ROUTINE IS NEEDED TO CALL RK2, AND A SUBROUTINE MUST BE AVAILABLE TO *
C COMPUTE THE SYSTEM DERIVATIVE.                                      *
C                                                                     *
C  ***PRECISION***                                                    *
C                                                                     *
C     ALL PARAMETERS AND VARIABLES ARE DEFINED AS DOUBLE PRECISION    *
C                                                                     *
C  ***ENVIRONMENT***                                                  *
C                                                                     *
C     VAX 11-780                                                      *
C                                                                     *
C**********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(N),YP(10),YHAT(10),YHATP(10)
      H=XEND-X
      CALL FCN(N,X,Y,YP)
      DO 10 I=1,N
         YHAT(I)=Y(I)+H*YP(I)
10    CONTINUE
      CALL FCN(N,XEND,YHAT,YHATP)
      DO 20 I=1,N
         Y(I)=0.5D0*(Y(I)+YHAT(I)+H*YHATP(I))
20    CONTINUE
      X=XEND
      RETURN
      END
```

```
      SUBROUTINE SIMQ(AD,A,B,N,ND,KS)
C
C*****************************************************************************
C                                                                            *
C     SUBROUTINE SIMQ IS AN OLD IBM SYSTEM USED TO SOLVE A SYSTEM OF         *
C SIMULTANEOUS LINEAR EQUATIONS.  THE ALGORITHM IS GAUSSIAN ELIMINATION.     *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  AD - MATRIX OF COUPLING COEFFICIENTS                                      *
C  A  - WORK SPACE MATRIX OF DIMENSION IDENTICAL TO THAT OF AD               *
C  B  - RIGHT HAND SIDE VECTOR                                               *
C  N  - RANK OF THE SYSTEM                                                   *
C  ND - NUMBER OF EQUATIONS IN THE SYSTEM (USUALLY EQUAL TO N)               *
C  KS - ERROR PARAMETER, KS=1 FOR A SINGULAR MATRIX                          *
C                                                                            *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION B(ND),AD(ND,ND),A(1)
      IJ=0
      DO 130 K=1,N                                                      ARRAY
      DO 130 L=1,N                                                      ARRAY
      IJ = IJ+1                                                         ARRAY
 130  A(IJ) = AD(L,K)                                                   ARRAY
 132  TOL=0.0                                                           SIMQ 540
      KS=0                                                              SIMQ 550
      JJ=-N                                                             SIMQ 560
      DO 65 J=1,N                                                       SIMQ 570
      JY=J+1                                                            SIMQ 580
      JJ=JJ+N+1                                                         SIMQ 590
      BIGA=0                                                            SIMQ 600
      IT=JJ-J                                                           SIMQ 610
      DO 30 I=J,N                                                       SIMQ 620
      IJ=IT+I                                                           SIMQ 660
      IF(DABS(BIGA)-DABS(A(IJ))) 20,30,30                               SIMQ 670
   20 BIGA=A(IJ)                                                        SIMQ 680
      IMAX=I                                                            SIMQ 690
   30 CONTINUE                                                          SIMQ 700
      IF(DABS(BIGA)-TOL) 35,35,40                                       SIMQ 740
   35 KS=1                                                              SIMQ 750
      GO TO 220                                                         SIMQ 751
   40 I1=J+N*(J-2)                                                      SIMQ 800
      IT=IMAX-J                                                         SIMQ 810
      DO 50 K=J,N                                                       SIMQ 820
      I1=I1+N                                                           SIMQ 830
      I2=I1+IT                                                          SIMQ 840
      SAVE=A(I1)                                                        SIMQ 850
      A(I1)=A(I2)                                                       SIMQ 860
      A(I2)=SAVE                                                        SIMQ 870
   50 A(I1)=A(I1)/BIGA                                                  SIMQ 910
      SAVE=B(IMAX)                                                      SIMQ 920
      B(IMAX)=B(J)                                                      SIMQ 930
      B(J)=SAVE/BIGA                                                    SIMQ 940
      IF(J-N) 55,70,55                                                  SIMQ 980
   55 IQS=N*(J-1)                                                       SIMQ 990
      DO 65 IX=JY,N                                                     SIMQ1000
      IXJ=IQS+IX                                                        SIMQ1010
      IT=J-IX                                                           SIMQ1020
      DO 60 JX=JY,N                                                     SIMQ1030
```

```
      IXJX=N*(JX-1)+IX                                    SIMQ1040
      JJX=IXJX+IT                                         SIMQ1050
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                     SIMQ1060
   65 B(IX)=B(IX)-(B(J)*A(IXJ))                           SIMQ1070
   70 NY=N-1                                              SIMQ1110
      IT=N*N                                              SIMQ1120
      DO 80 J=1,NY                                        SIMQ1130
      IA=IT-J                                             SIMQ1140
      IB=N-J                                              SIMQ1150
      IC=N                                                SIMQ1160
      DO 80 K=1,J                                         SIMQ1170
      B(IB)=B(IB)-A(IA)*B(IC)                             SIMQ1180
      IA=IA-N                                             SIMQ1190
   80 IC=IC-1                                             SIMQ1200
  220 IF (N.EQ.ND) RETURN                                 ARRAY
      IJ = N*N+1                                          ARRAY
      DO 110 L=1,N                                        ARRAY
      DO 110 K=1,N                                        ARRAY
      IJ = IJ-1                                           ARRAY
  110 AD(N-L+1,N-K+1) = A(IJ)                             ARRAY
      RETURN                                              SIMQ1210
      END                                                 SIMQ1220
```

```
      FUNCTION SIMS(F,ZL,ZU,NINT)
C***********************************************************************
C                                                                      *
C    THIS ROUTINE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS *
C AND ACOUSTICS, STANFORD UNIVERSITY BY THOMAS LUND.  LATEST REVISION *
C 18 JULY 1986.                                                        *
C                                                                      *
C    SUBROUTINE SIMS PERFORMS A NUMERICAL INTEGRATION OF A ONE DI-     *
C MENSIONAL FUNCTION USING A FOURTH ORDER ACCURATE SCHEME.             *
C                                                                      *
C  **PARAMETER DESCRIPTION**                                           *
C                                                                      *
C    INPUTS:                                                           *
C   F     - A FUNCTION WHICH DEFINES THE EQUATION TO BE INTEGRATED.    *
C           THE CALL MUST BE OF THE FORM CALL F(Z) WHERE Z IS          *
C           VARIABLE OF INTEGRATION. F MUST BE DECLARED EXTERNAL IN    *
C           THE DRIVING ROUTINE                                        *
C   ZU  -   UPPER LIMIT OF INTEGRATION.                                *
C   ZL  -   LOWER LIMIT OF INTEGRATION.                                *
C   NINT - NUMBER OF INTEGRATION SUBINTERVALS.  NINT MUST BE AN EVEN  *
C           NUMBER.                                                    *
C                                                                      *
C  **LINKING** - NO LINKING TO LIBRARIES NEEDED, HOWEVER A DRIVING     *
C                ROUTINE AS WELL AS A FUNCTION EVALUATING ROUTINE      *
C                NEED TO BE SUPPLIED.                                  *
C                                                                      *
C  **PRECISION** - ALL PARAMETERS AND INTERNAL VARIABLES ARE DOUBLE    *
C                  PRECISION.                                          *
C                                                                      *
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C        *** COMPUTE SUB-INTERVAL SIZE ***
C
      DZ=(ZU-ZL)/DFLOAT(NINT)
C
C        *** INITIALIZE INTEGRATION PARAMETER ***
C
      SUM=0.0D0
      Z=ZL
C
C        *** CARRY OUT INTEGRATION USING 4TH ORDER SIMPSONS RULE ***
C
      DO 10, I=0,NINT
C
C           *** SET WEIGHTING FACTOR TO A NOMINAL VALUE OF 2 ***
C
        R=2.0D0
C
C           *** IF I IS ODD SET THE WEIGHTING FACTOR TO 4 ***
C
        IF((DFLOAT(I)/2.0D0-DFLOAT(I/2)).GT.0.2D0) R=4.0D0
C
C           *** AT THE ENDPOINTS SET THE WEIGHTING FACTOR TO 1 ***
C
        IF(I.EQ.0.OR.I.EQ.NINT) R=1.0D0
C
C           *** FIND CONTRIBUTION TO THE INTEGRAL ***
C
        SUM=SUM+R*F(Z)
```

```
         Z=Z+DZ
10    CONTINUE
C
C        *** RETURN THE APPROXIMATED INTEGRAL ***
C
      SIMS=SUM*DZ/3.0D0
C
      RETURN
      END
```

Bibliography

[1] Porter J. L. and Squyers R. A. *A Summary/Overview of Ejector Augmentor Theory and Performance.* Technical Report ATC Report No. R-91100/9cr-47A, Office of Naval Research and The Air Force Office of Scientific Research, 1979.

[2] Quinn B. *Thrust Augmenting Ejector: A Review of the Application of Jet Mechanics to V/STOL Aircraft Propulsion.* Technical Report CPP-308, AGARD, 1981.

[3] von Karman T. Theoretical remarks on thrust augmentation. In *Contributions to Applied Mechanisc, Reissner Anniversary Volume*, pages 461–468, Edwards, J. W., Ann Arbor Mich., 1949.

[4] Keenan J. H., Neumann E. P., and Lustwerk F. An investigation of ejector design by analysis and experiment. *Journal of Applied Mechanics*, 17(3):299–309, 1950.

[5] Quinn B. *Wind Tunnel Investigation of the Forces Acting on an Ejector in Flight.* Technical Report 70-0141, Aerospace Research Laboratories, July 1970.

[6] Nagaraja K. S., Hammond D. L., and Graetch J. E. One-dimensional compressible ejector flows. AIAA Paper 73-1184, 1973.

[7] Hedges K. R. and Hill P. G. Compressible flow ejectors - development of a finite difference flow model. *Transactions of the AMSE, Ser. I. J. Fluids Engineering*, 96(3):272–288, 1974.

[8] Lasinski T. A., Andrews A. E., Sorenson R. L., Chaussee D. S., Pulliam T. H., and Kutler P. Computation of the steady viscous flow over a tri-element augmentor-wing airfoil. AIAA Paper 82-0021, 1982.

[9] Bevilaqua P. M. and McCullough J. K. Entrainment method for v/stol ejector analysis. AIAA Paper 76-419, 1976.

[10] Bevilaqua P. M. and DeJoode A. D. *Viscid/Inviscid Interaction Analysis of thrust Augmenting Ejectors.* Technical Report CP 2093, NASA, 1979.

[11] Bevilaqua P. M., Woan C. J., and Schum E. F. *Viscous/Inviscid Interaction Analysis of Ejector Wings.* Technical Report AFWAL-TR-82-3059, Air Force Wright Aeronautical Laboratories, 1982. Proceedings: Ejector Workshop For Aerospace Applications.

[12] Tavella D. A. and Roberts L. *Analysis of Confined Turbulent Jets.* Technical Report JIAA TR-51, Joint Institute for Aeronautics and Acoustics, Stanford University, December 1982.

[13] Tavella D. A. and Roberts L. A simple viscous-inviscid aerodynamic analysis of two-dimensional ejectors. AIAA Paper 84-0281, 1984.

[14] Lund T. S., Tavella D. A., and Roberts L. A zonal computational approach to thrust augmentor optimization. AIAA Paper 85-0110, 1985.

[15] Foa J. V. *Energy Transfer Rate In Axial-Flow Crypto-Steady Pressure Exchange.* Technical Report TR AE 6102, Rensselaer Polytechnic Institute, Troy, New York, February 1961.

[16] Hohenemser K. H. and Porter J. L. Contribution to the theory of rotary jet flow induction. *Journal of Aircraft*, 3(4):339–346, July-August 1966.

[17] Viets H. Flip-flop jet nozzle. *AIAA Journal*, 13(10):1375–1379, October 1975.

[18] Stow P. Incorporation of viscous-inviscid interactions in turbomachinery design. In *Advanced Study Institute on Thermodynamics and Fluid Mechanics of Turbomachinery*, NATO, September 1985.

[19] Hirsch C. and Janssens P. Viscous-inviscid interactions in cascades. *Israel Journal of Technology*, 20(3):109–126, 1982.

[20] Streett C. L. Viscous-inviscid interaction method including wake effects for three-dimensional wing-body configurations. AIAA Paper-81-1266, 1984.

[21] Waggoner E. G. Transonic three-dimensional viscous-inviscid interaction for wing-body configuration analysis. AIAA Paper 82-0163, 1982.

[22] Baker A. J. and Orzechowski J. A. A viscous-inviscid interaction algorithm for three-dimensional turbulent subsonic aerodynamic juncture region flow. AIAA Paper 82-0100, 1982.

[23] Inger G. R. and Habashi W. G. Modular application of a shock/boundary layer interaction solution to supercritical viscous-inviscid flow field analysis. In *Computational Methods in Viscous Flows*, pages 1–38, Pinridge Press, Swansea, England, 1984.

[24] Edwards D. E., Carter J. E., and Hafez M. M. Viscous/inviscid interaction analysis of transonic shock induced separated flow including normal pressure gradients. AIAA Paper 85-0371, 1985.

[25] Liou M. S. Analysis of viscous-inviscid interactions in transonic internal flows with a shock wave and mass transfer. AIAA Paper 81-0004, 1981.

[26] Meier H. U. and Kreplin H. P. Boundary layer separation due to weak and strong viscous-inviscid interaction on an inclined body of revolution. In *Viscous and Interacting Flow Field Effects*, Air Force Wright Aeronautical Laboratories, 1984. Report Number AFWAL-AD-P004789.

[27] Pletcher R. H. Calculation of separated flows by viscous-inviscid interaction. In *Computational Methods in Viscous Flows*, pages 365–394, Pinridge Press, Swansea, England, 1984.

[28] van Dalsem W. R. *Simulation of Separated Transonic Airfoil Flow by Finite-Difference Viscous-Inviscid Interaction*. PhD thesis, Stanford University, 1984.

[29] Kwon O. K. Application of a viscous-inviscid interaction procedure to predict turbulent separating flows over a rearward-facing step. In *Computation of Internal Flows: Methods and Applications; Proceedings of the Energy Sources Technology Conference*, pages 65–71, American Society of Mechanical Engineers, AMSE Press, New York, 1983.

[30] Williams B. R. The prediction of separated flow using a viscous-inviscid interaction method. In *International Council of the Aeronautical Sciences*, American Institute of Aeronautics and Astronautics, New York, 1984.

[31] Hall E. J. and Pletcher R. H. Application of a viscous-inviscid interaction procedure to predict separated flows with heat transfer. ASME Paper 83-WA/HT-7, 1984.

[32] Vatsa V. N. and Verdon J. M. Viscous/inviscid interaction analysis of asymmetric trailing-edge flows. AIAA Paper 84-0266, 1984.

[33] Le Balleur J. C. and Girodroux-Lavigne P. A semi-implicit and unsteady numerical method of viscous-inviscid interaction for transonic separated flows. *La Recherche Aerospatiale*, ISSN 0379-380x(1):15–37, 1984.

[34] Inger G. R. and Huang M. K. Application of unsteady laminar triple-deck theory to viscous-inviscid interactions form an oscillating flap in supersonic and subsonic flow. *Journal de Mechanique Theorique et Appliquee*, 2(3):325–349, 1983.

[35] McCroskey W. J. and Pucci S. L. Viscous-inviscid interactions on oscillating airfoils in subsonic flow. AIAA Paper 81-0051, 1981.

[36] Lund T. S. *A Zonal Computational Procedure Adapted to the Optimization of Two-Dimensional Thrust Augmentor Inlets.* Technical Report JIAA TR-6, Joint Institute for Aeronautics and Acoustics, Stanford University, July 1985.

[37] Churchill R. V. and Brown J. W. *Complex Variables and Applications.* McGraw-Hill, New York, 1984.

[38] Milne-Thomson L. M. *Theoretical Hydrodynamics.* Macmillan Press, London, 1972.

[39] Holst T. L. *Numerical Computation of Transonic Flow Governed by the Full Potential Equation.* Technical Report TM-84310, NASA, 1983.

[40] Hess J. L. and Smith A. M. O. Calculation of potential flow about arbitrary bodies. In *Progress in Aeronautical Sciences*, Pergamon Press, 1967.

[41] Ashley H. and Landahl M. *Aerodynamics of Wings and Bodies.* Addison-Wesley, Reading Mass., 1965.

[42] Kuetue A. M. and Chow C. Y. *Foundations of Aerodynamics: Bases of Aerodynamic Design.* John Wiley and Sons, New York, 1976.

[43] Carrier G. F. and Pearson C. E. *Partial Differential Equations.* Academic Press, New York, 1976.

[44] Hess J. L. and Friedman D. M. Analysis of complex inlet configurations using a higher-order panel method. AIAA Paper 83-1828, 1983.

[45] Hess J. L. Higher order numerical solution of the integral equation for the two-dimensional neumann problem. *Computer Methods in Applied Mechanics and Engineering*, 2:1–15, 1973.

[46] Shenk A. *Calculus and Analytic Geometry.* Goodyear Publishing Co., Santa Monica, CA, 1977.

[47] Bernal L. and Sarohia V. Entrainment and mixing in thrust augmenting ejectors. AIAA Paper 83-0172, 1983. see also JPL report no. JPL D-66.

[48] Abramovich G. N. *The Theory of Turbulent Jets.* M.I.T. Press, Cambridge, Mass., 1963.

[49] Lund T. S., Tavella D. A., and Roberts L. Analysis of interacting dual lifting ejector systems. AIAA Paper 86-0478, 1986.

[50] Tollmien W. Berechnung turbulenter ausbreitungsvorgange. *Z.A.M.M.*, 6:468–478, 1926. English Translation N.A.C.A. TM 1085, 1945.

[51] Gortler H. Berechnung von aufgaben der freien turbulenz auf grund eines neuen naherungsansatzes. *Z.A.M.M.*, 22:244–254, 1942.

[52] Newmann B. G. Turbulent wakes and jets in a pressure gradient. In Gino Sovran, editor, *Fluid Mechanics of Internal Flows*, Elsevier Publishing Co., 1967.

[53] Bevilaqua P. M. Lifting surface theory for thrust-augmenting ejectors. *AIAA Journal*, 16(5):475–481, 1987.

[54] Quinn B. A simple estimate of the effect of ejector length on thrust augmentation. *Journal of Aircraft*, 10(5):313–314, 1973.

[55] Anderson D. A., Tannehill J. C., and Pletcher R. H. *Computational Fluid Mechanics and Heat Transfer.* McGraw Hill, New York, 1984.

[56] Patankar S. V. *Numerical Heat Transfer and Fluid Flow.* Hemisphere, Washington D. C., 1980.

[57] Croft D. R. and Lilley D. G. Finite difference performance analysis of jet pumps. *AIAA journal*, 14(10), 1976.

[58] Gilbert G. B. and Hill P. G. *Analysis and Testing of Two Dimensional Slot Nozzle Ejectors with Variable Area Mixing Sections.* Technical Report CR-2251, NASA, 1973.

[59] von Karman T. Uber laminare und tubulente reibung. *Z.A.M.M.*, 1:233–252, 1921. English Translation NACA report no. 1092.

[60] Pohlhausen K. Naherungsweisen integration der differentialgleichung der laminaren reibungsschicht. *Z.A.M.M.*, 1:252–268, 1921.

[61] Curtet R. Confined jets and recirculating phenomena with cold air. *Combustion and Flame*, 2, 1958.

[62] Hill P. Turbulent jets in ducted streams. *Journal of Fluid Mechanics*, 22, 1965. Part 1.

[63] Finlayson B. A. *The Method of Weighted Residuals and Variational Principles.* Academic Press, New York, 1972.

[64] Rajaratnam N. *Turbulent Jets.* Elsevier Scientific Publishing Co., New York, 1976.

[65] Gill P. E., Murray W., and Pitfield. *The Implementation of two Revised Quasi-Newton Algorithms for Unconstrained Optimization.* Technical Report DNAC 11, National Physics Laboratory, Division of Numerical Analysis and Computing, 1972.

[66] Gill P. E., Murray W., and Wright M. H. *Practical Optimization.* Academic Press, London, 1981.

[67] Alperin M. and Wu J-J. *Jet-Diffuser Ejector - Attached Nozzle Design.* Technical Report CR 152361, NASA, 1980.